Freescale Semiconductor
Application Note

Document Number: AN4445
Rev. 1, 8/2012

Kinetis 100 MHz Rev 1.x to Rev 2.x Migration Guide

by:
MSG IMM Applications
Austin, Texas

Contents

1 Purpose and overview

This document shows differences and describes the details of migrating from Kinetis 100 MHz Rev. 1.x to Rev. 2.x.

1.1 Part numbering and mask set information

Revision	Mask Set	Part Number Example
1.0	0M33Z	PK10N512VMD100
1.1	0N30D	N/A
1.2	1N30D / 2N30D (functionally identical)	PK10N512VMD100
1.4	4N30D	MK10DN512ZVMD10 (‘Z’ character: INITIAL Production mask set)
2.2	2N22D	MK10DN512VMD10 (no ‘Z’ character: PRODUCTION mask set)

1.2 About this document

This document is divided up into three major sections—Improvements, Updated modules, and Enhanced modules.

The Improvements section outlines the various changes that have been implemented and that require no changes in either software or hardware.

The Updated modules section outlines the modules that have been updated to use newer versions. The overall functionality provided is similar; however, changes are required in software. Hardware changes may be required in order to use new features.

Enhanced modules section outlines the existing modules that have undergone minor changes. Software and hardware changes may be required in order to use new features.

Additionally, a color coding scheme has been used throughout this document where:

- GREEN: Designates new additions,
- YELLOW: Designates changes, and
- RED: Designates removals

2 Improvements

2.1 Low-power improvements

Change	Hardware impact	Software impact
Reduces power consumption in dynamic power modes • ~10% power reduction in Run and Wait modes due by optimizing clocking architecture • ~3 mA baseline power reduction in Run and Wait modes due to new flash analog circuitry	If battery operated, expect longer life.	None
Faster VLLSx wakeup times • ~10% faster VLLSx wakeup times due to improvements in recovery procedure	If battery operated, expect longer life.	None
Support for VLPR and VLPW modes • < 2 mA current consumption up to 4 MHz core and bus clocks (flash max frequency is 1 MHz) with fully functional peripherals	If battery operated, expect longer life.	No software impact, unless modes are added to application. To configure modes refer to the SMC module.
Increases core frequency in VLPR/ VLPW modes • Maximum VLPR and VLPW core frequency changes from 2 MHz to 4 MHz with 1:1 core to bus clock ratio support improving peripheral performance with ultra-low-power current consumption	None	No software impact, unless modes are added to application. To change from default 2 MHz to faster 4 MHz faster IRC output in VLPR/VLPW, configure the MCG_SC[FCRDIV] register field to divide factor of 1.

Table continues on the next page...

Change	Hardware impact	Software impact
Support for disabling Flash in dynamic power modes for SRAM execution • Removes Flash power consumption overhead most notable in <10 MHz core frequencies; allowing code execution from SRAM • Supports temporary doze of Flash during WAIT mode with automatic Flash start-up upon return to Run mode.	If battery operated, expect longer life.	No software impact, unless disabling of flash is added to application. To disable or doze flash configure the SIM_FCFG1[FLASHDIS or FLASHDOZE] register bits.

2.2 Flash timings

The following flash memory parameters in the data sheet are improved by ~ 3 times in Rev. 2.x:

- tersblk256k
- tersall
- tpgmpart256k

3 Updated Modules

3.1 Power Management Controller (PMC)

The primary duties of the Power Management Controller (PMC) are: control the regulator, control the POR and LVD circuits, and provide voltage and current sources for the MCU. For Kinetis 100 MHz Rev. 2.x120 MHz devices the I/O retention control is now in the PMC. The functionality of the PMC is linked closely with the Low Leakage Wakeup Unit (LLWU), the System Mode Controller (SMC), and the new Reset Control Module (RCM).

3.1.1 Memory map comparison

The register map and names have remained the same. The figure below shows the only bit level changes to the memory map.

Bit	7	6	5	4	3	2	1	0
Read	0			TRAMPO	VLPRS	REGONS		BGBE
Write							0	
Reset	0	0	0	0	0	1	0	0
Bit	7	6	5	4	3	2	1	0
Read	0			BGEN	ACKISO	REGONS	0	BGBE
Write					w1c			
Reset	0	0	0	0	0	1	0	0

Figure 1. PMC_REGSC

Changed bit/field names:

- TRAMPO—Removed in Kinetis 100 MHz Rev. 2.x120 MHz
- VLPRS—Removed in Kinetis 100 MHz Rev. 2.x120 MHz
- BGBE—Bandgap Buffer Enable

3.1.2 Software impact

The PMC register names are the same; however, some important bit changes have been made.

TRAMPO: The TRAMPO bit has been removed. There is a new bit in the SMC—RAM2PO—that controls the RAM power in VLLS2. The new RAM2PO bit has different functionality, however.

VLPRS: The VLPRS bit has been removed. Software only needs to check the Run Regulator Status and the SMC_PMSTAT register when entering VLPR mode. The REGONS bit will clear when the regulator is in stop regulation or transition to/from it and the SMC_PMSTAT will read 04 when the MCU power mode is in VLPR.

Bandgap Enable: This new bit controls the enable of the bandgap circuit. If the VREF module is operational, the BGPE bit should be set. Note that if the BGBE is set, the bandgap circuit will remain enabled and will consume current in all of the power modes. To keep the current to a minimum in the lowest power modes, this bit should be cleared prior to mode entry.

ACKISO: The PMC has the acknowledge to clear the hold on the state of the I/O pins and oscillator module. This bit used to reside in the LLWU. This bit must be cleared upon recovery from VLLSx modes to release the hold on the I/O and oscillator modules.

3.1.3 Hardware impact

Bandgap Enable (BGEN): The new bit has hardware interaction impact. If BGEN is set, then the bandgap is turned on. The operation of this bit may be needed by other circuitry like the VREF module or ADC.

ACKISO: The ACKISO bit is an important bit to consider when working with the low power modes of operation. The MCU recovers from VLLSx through the reset flow. Typically the port I/O, module initialization of timers, communications modules, and the oscillator should be initialized prior to acknowledging the release of the I/O.

3.2 Mode Controller Version 1 to Version 2

The primary duty of the Mode Controller (MC) and System Mode Controller (SMC) has been to control the entry into and exit from each power mode. The module has a new name in Revision 2.0. The MC now is the SMC. The functionality of the SMC has always been linked closely with the Power Management Controller (PMC), the Low Leakage Wakeup Unit (LLWU), and the new Reset Control Module (RCM).

3.2.1 Memory Map Comparison

There are two new registers in the SMC module. They are SMC_VLLSCTRL and SMC_PMSTAT. The two SRS registers that are in the MC have been moved to the RCM. The changes to this register will be discussed in the RCM section.

Address: MC_PMPROT is 4007_E000h base + 2h offset = 4007_E002h

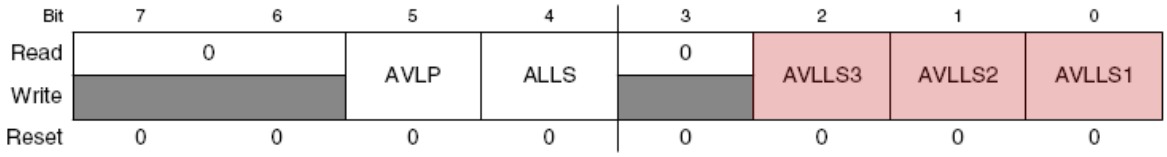


Figure 2. MC_PMPROT

Address: SMC_PMPROT is 4007_E000h base + 0h offset = 4007_E000h

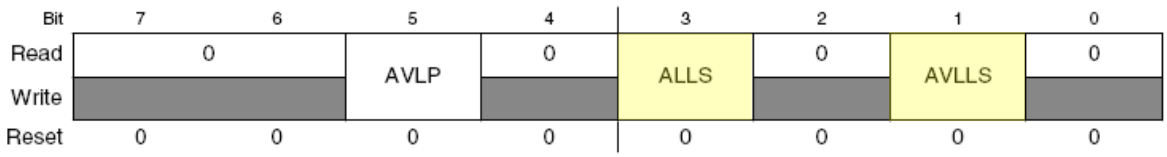


Figure 3. SMC_PMPROT

Changed bit/field names:

- ALLS—Moved to bit 3
- AVLLS3—AVLLS
- AVLLS2—AVLLS
- AVLLS1—AVLLS

Address: MC_PMCTRL is 4007_E000h base + 3h offset = 4007_E003h

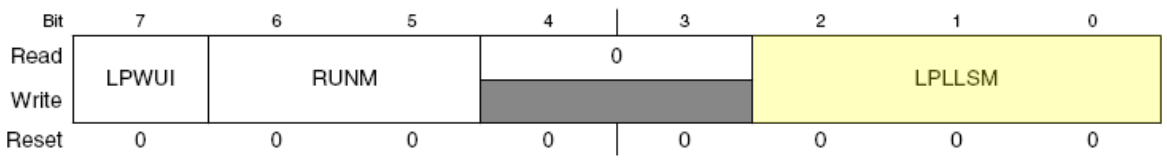


Figure 4. MC_PMCTRL

Address: SMC_PMCTRL is 4007_E000h base + 1h offset = 4007_E001h

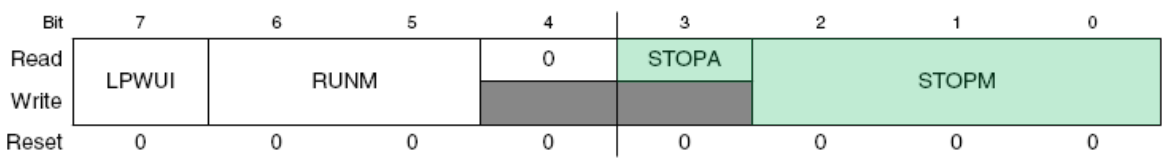


Figure 5. SMC_PMCTRL

Changed bit/field names:

- STOPA—New bit in the SMC
- LPLLSM—VLLSM, moves to new register SMC_VLLSSCTRL
- STOPM—New bit field in the SMC

Address: SMC_VLLSCTRL is 4007_E000h base + 2h offset = 4007_E002h

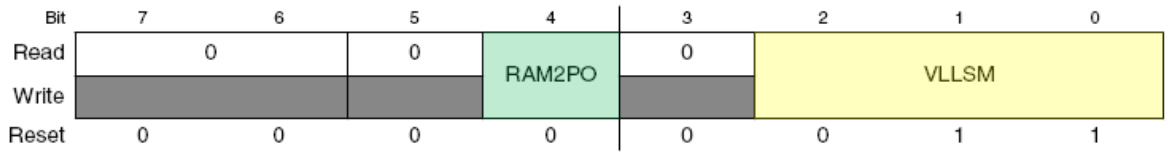


Figure 6. SMC_VLLCTRL

Changed bit/field names:

- RAM2PO—New bit in the SMC
- VLLSM—New bit field in the SMC

Address: SMC_PMSTAT is 4007_E000h base + 3h offset = 4007_E003h

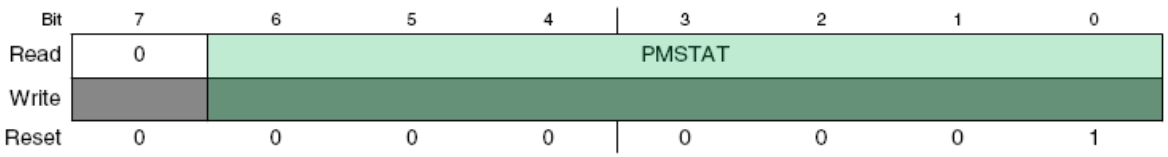


Figure 7. SCM_PMSTAT

Changed bit/field names:

PMSTAT—New bit field in the SMC

3.2.2 Software impact

General impact: Any register reference to the MC register will need to change the RCM and SMC register names. The SRS registers are in the RCM. The prefix for the mode control functions changes from MC to SMC.

SMC_PMPROT: The SMC_PMPROT register is still a one-time-write after reset type of register. The protection of entry into the VLLSx low-power modes is controlled by one bit in the SMC. Software will need to write to the PMPROT in the initialization code to allow only the desired low-power modes. The address changes of the registers and the location change of the ALLS bit is handled by the header file definitions for the SMC.

SMC_PMCTRL and SMC_VLLSCTRL:

The SMC_PMCTRL and SMC_VLLSCTRL register control mode entry and exit. Software will need to change the values written to these registers to accomplish the functions done in the MC_PMCTRL register.

A read of the new bit STOPA in the SMC_PMCTRL register will indicate that an interrupt or reset occurred during a stop entry sequence and the SMC can abort the transition early and return to Run mode without completely entering the stop mode. See the reference manual SMC functional description for more information.

The new STOPM bit field has the following meaning:

2–0 STOPM	Stop Mode Control When written, this field controls entry into the selected stop mode when sleep-now or sleep-on-exit mode is entered with SLEEPDEEP=1 . When this field is set to VLLSx, the VLLSCTRL register is used to further select the particular VLLS sub-mode which will be entered. Writes to this field are blocked if the protection level has not been enabled using the PMPROT register. After any system reset, this field is cleared by hardware on any successful write to the PMPROT register. 000 Normal stop (STOP) 001 Reserved 010 Very low power stop (VLPS) 011 Low leakage stop (LLS) 100 Very low leakage stop (VLLSx) 101 Reserved 110 Reseved 111 Reserved

Low-Power Mode Entry Change: If a low-power mode of Stop, VLPS, or LLS is desired, the PMPROT bit must first be set and the a write to STOPM is needed. If one of the VLLSx modes is desired, one additional write to the VLLSM bits in the SMC_VLLSCTRL is needed.

2–0 VLLSM	VLLS Mode Control This field controls which VLLS sub-mode to enter if STOPM=VLLS. 000 Reserved 001 VLLS1 010 VLLS2 011 VLLS3 100 Reserved 101 Reserved 110 Reserved 111 Reserved

SMC_PMSTAT

A read of this new register in the SMC will give you the current low-power mode.

3.2.3 Hardware impact

RAM power in VLLS2 mode

A new bit in the SCM_VLLSCTRL register—RAM2PO—controls whether the RAM partition 2 is powered or not while in VLLS2 low-power mode.

VLPR and VLPW now supported

With the SMC the mode VLPR and VLPW are now supported. The way you enter the modes is the same between revisions.

Debug in low-power modes

Please note that the SMC allows debug operation in Run, Wait, VLPR, or VLPW in the same way as in the MC. The debugger handles attempts to enter Stop and VLPS by entering an emulated stop state. See the reference manual SMC functional description for more information.

3.3 Reset Controller Module (RCM)

The RCM is new on Rev. 2; however, not all of the functions that are in this module are new. The SRS registers that were in the MC in Rev. 1 are now in the RCM. The SRS register has been revised some.

3.3.1 Memory map comparison

Address: MC_SRSL is 4007_E000h base + 1h offset = 4007_E001h

Bit	7	6	5	4	3	2	1	0
Read	POR	PIN	COP	0		LOC	LVD	WAKEUP
Write								
Reset	1	0	0	0	0	0	1	0

Address: MC_SRSH is 4007_E000h base + 0h offset = 4007_E000h

Bit	7	6	5	4	3	2	1	0
Read	0					SW	LOCKUP	JTAG
Write								
Reset	0	0	0	0	0	0	0	0

Figure 8. SRS Register High and Low (MC_SRSH)

Address: RCM_SRS0 is 4007_F000h base + 0h offset = 4007_F000h

Bit	7	6	5	4	3	2	1	0
Read	POR	PIN	WDOG	0		LOC	LVD	WAKEUP
Write								
Reset	1	0	0	0	0	0	1	0

Figure 9. SRS Register 0 (RCM_SRS0)

Address: RCM_SRS1 is 4007_F000h base + 1h offset = 4007_F001h

Bit	7	6	5	4	3	2	1	0
Read	0	0	SACKERR	EZPT	MDM_AP	SW	LOCKUP	JTAG
Write								
Reset	0	0	0	0	0	0	0	0

Figure 10. SRS Register 1 (RCM_SRS1)

Changed bit/field names:

- LOL → Bit 3 in RCM_SRS0 - New bit field in Rev. 2.x
- SACKERR → Bit 5 in RCM_SRS1 - New bit field in Rev. 2.x
- EZPT → Bit 4 in RCM_SRS1 - New bit field in Rev. 2.x
- MDM_AP → Bit 3 in RCM_SRS1 - New bit field in Rev. 2.x

Address: RCM_RPFC is 4007_F000h base + 4h offset = 4007_F004h

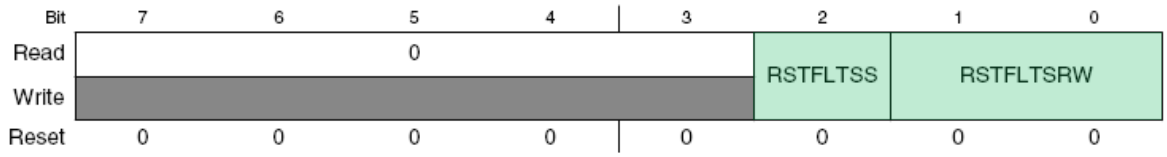


Figure 11. Reset Pin Filter Control Register (RCM_RPFC)—New in Rev. 2.x

Address: RCM_RPFW is 4007_F000h base + 5h offset = 4007_F005h

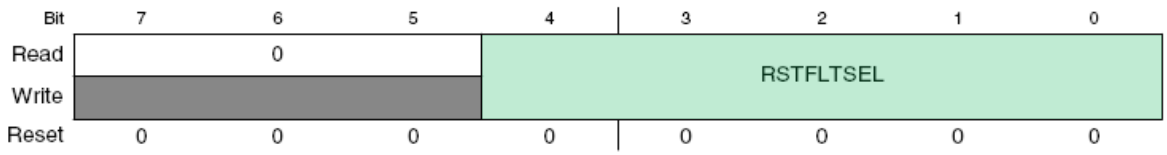


Figure 12. Reset Pin Filter Width Register (RCM_RFPW)—New in Rev. 2.x

3.3.2 Software impact

Any references to MC_SRSH and MC_SRSL will need to be changed to the new register names. Software that decoded the reset information in these two registers will need to account for the new bits added in each of the two registers.

If you want to use the new digital filter function of the RCM, please refer to the reference manual sections Reset and Boot and Reset Control Module.

3.3.3 Hardware impact

There are four addition sources of reset that were identified in Rev. 2.

The Reset pin input can be digitally filtered with either the LPO clock or the bus clock. The bus clock filter value is selectable in the RCM_RFPW register.

3.4 LLWU

The Low Leakage Wakeup Unit (LLWU) controls the exit from LLS and VLLSx power modes. The functionality of the LLWU is linked closely with the Power Management Controller (PMC), the System Mode Controller (SMC), and the new Reset Control Module (RCM).

3.4.1 Memory map comparison

There are three new registers in the LLWU module. They are the LLWU_FILT1, LLWU_FILT2, and LLWU_RST registers. One register has been removed in Rev. 2, the LLWU_CS.

Address: LLWU_ME is 4007_C000h base + 4h offset = 4007_C004h

Bit	7	6	5	4	3	2	1	0
Read	WUME7	WUME6	WUME5	WUME4	WUME3	WUME2	WUME1	WUME0
Write								
Reset	0	0	0	0	0	0	0	0

Figure 13. LLWU_ME Register

Changed bit/field names:

WUME7—Connected to RTC Seconds enable

Address: LLWU_F3 is 4007_C000h base + 7h offset = 4007_C007h

Bit	7	6	5	4	3	2	1	0
Read	MWUF7	MWUF6	MWUF5	MWUF4	MWUF3	MWUF2	MWUF1	MWUF0
Write								
Reset	0	0	0	0	0	0	0	0

Figure 14. LLWU_F3 Register

Changed bit/field names:

WUMF7—Connected to RTC Seconds interrupt

The RTC seconds flag is connected to bit 7 of the corresponding LLWU flag register.

Address: LLWU_CS is 4007_C000h base + 8h offset = 4007_C008h

Bit	7	6	5	4	3	2	1	0
Read	ACKISO	0					FLTEP	FLTR
Write	w1c					1		
Reset	0	0	0	0	0	1	0	0

Figure 15. LLWU_CS—Rev. 1 removed

Changed bit/field names:

- ACKISO—Moved to bit 3 of the PMC_REGSC register
- FLTEP—Filter functionality changed

Address: LLWU_FILT1 is 4007_C000h base + 8h offset = 4007_C008h

Bit	7	6	5	4	3	2	1	0
Read	FILTF	FILTE		0	FILTSEL			
Write	w1c							
Reset	0	0	0	0	0	0	0	0

Figure 16. LLWU_FILT1—New register Rev. 2.x

Address: LLWU_FILT2 is 4007_C000h base + 9h offset = 4007_C009h

Bit	7	6	5	4	3	2	1	0
Read	FILTF	FILTE		0	FILTSEL			
Write	w1c							
Reset	0	0	0	0	0	0	0	0

Figure 17. LLWU_FILT2—New register Rev. 2.x

Address: LLWU_RST is 4007_C000h base + Ah offset = 4007_C00Ah

Bit	7	6	5	4	3	2	1	0
Read	0						LLRSTE	RSTFILT
Write								
Reset	0	0	0	0	0	0	1	0

Figure 18. LLWU_RST—New register Rev. 2.x

The functionality of the LLRSTE and RSTFILT bits is enabled when there is no dedicated Reset pin. Kinetis parts have a dedicated Reset pin; therefore, these bits do not change the operation of the Reset pin.

3.4.2 Software impact

LLWU Module Enable and Flag

Not obvious in the register description for Rev. 2.x is the fact that the RTC Seconds module interrupt is connected to bit 7 of the LLWU_ME register [WUME7]. The RTC seconds flag is connected to bit 7 [MWUF7] of the corresponding LLWU flag register LLWU_F3.

LLWU pin and reset filters

Register LLWU_FILT1, LLWU_FILT2, and LLWU_RST control the filter functionality of the LLWU. See the reference manual LLWU register definition and functional description for more information.

ACKISO move

The ACKISO bit is now in the PMC module registers. Software will need to re-target the write-1-to-clear operation to the new register. This is an important bit for users who are waking up from VLLSx modes through the reset flow. A read of the ACKISO bit indicates whether the I/O pads and the system oscillator(s) are in a latched state. Care should be used in not clearing this bit too early. You should re-initialize the I/O and the oscillator before writing the ACKISO or there might be a glitch on the pins or in the oscillator startup.

3.4.3 Hardware impact

LLWU_M7IF → RTC seconds: In Rev. 2, Module Wakeup bit 7 is connected to the output of the RTC seconds interrupt. This allows the RTC seconds output to wake up the MCU from LLS and VLLSx low-power modes.

LLWU pin filter function: The functionality of the pin filter in the LLWU module has changed. In Rev. 1.x, all of the LLWU inputs are fed into one pin filter circuitry resulting in a single wakeup flag. In Rev. 2.x, only two preselected LLWU inputs are fed into two pin filter circuits resulting in two possible filtered pin wakeup flags.

The LLWU in Rev. 2.x has filter and pin enable for the reset pin to wake up the MCU from LLS and VLLSx low power modes. For devices like Kinetis that have a dedicated Reset pin, the wakeup from LLS and VLLSx modes is always enabled and these bits do not enable or disable the Reset pin as a wakeup source in low-power modes.

3.5 RNG-B to RNG-A

Kinetis 100 MHz 1.x versions use random number generator version B (RNG-B) while the latest versions of the Kinetis family use random number generator version A (RNG-A). While these two versions are radically different, the migration is surprisingly simple. This section details the differences between the two versions and describes the necessary changes to your Kinetis setup to ensure a smooth transition from RNG-B to RNG-A.

3.5.1 RNG-A vs. RNG-B

The RNG-B variant is a cryptographically strong random number generator with three distinctive features:

- National Institute of Standards and Technology (NIST)-approved pseudo-random number generator (http://csrc.nist.gov)
- Inclusion of the key generation algorithm defined in the Digital Signature Standard (http://www.itl.nist.gov/fipspubs/fip186.htm)
- Integrated entropy sources capable of providing the PRNG with entropy for its seed

RNG-B uses a true random number generator module (TRNG) to add entropy to a register from which the pseudo-random number generator is seeded. See below for a detailed block diagram of the RNG-B module.

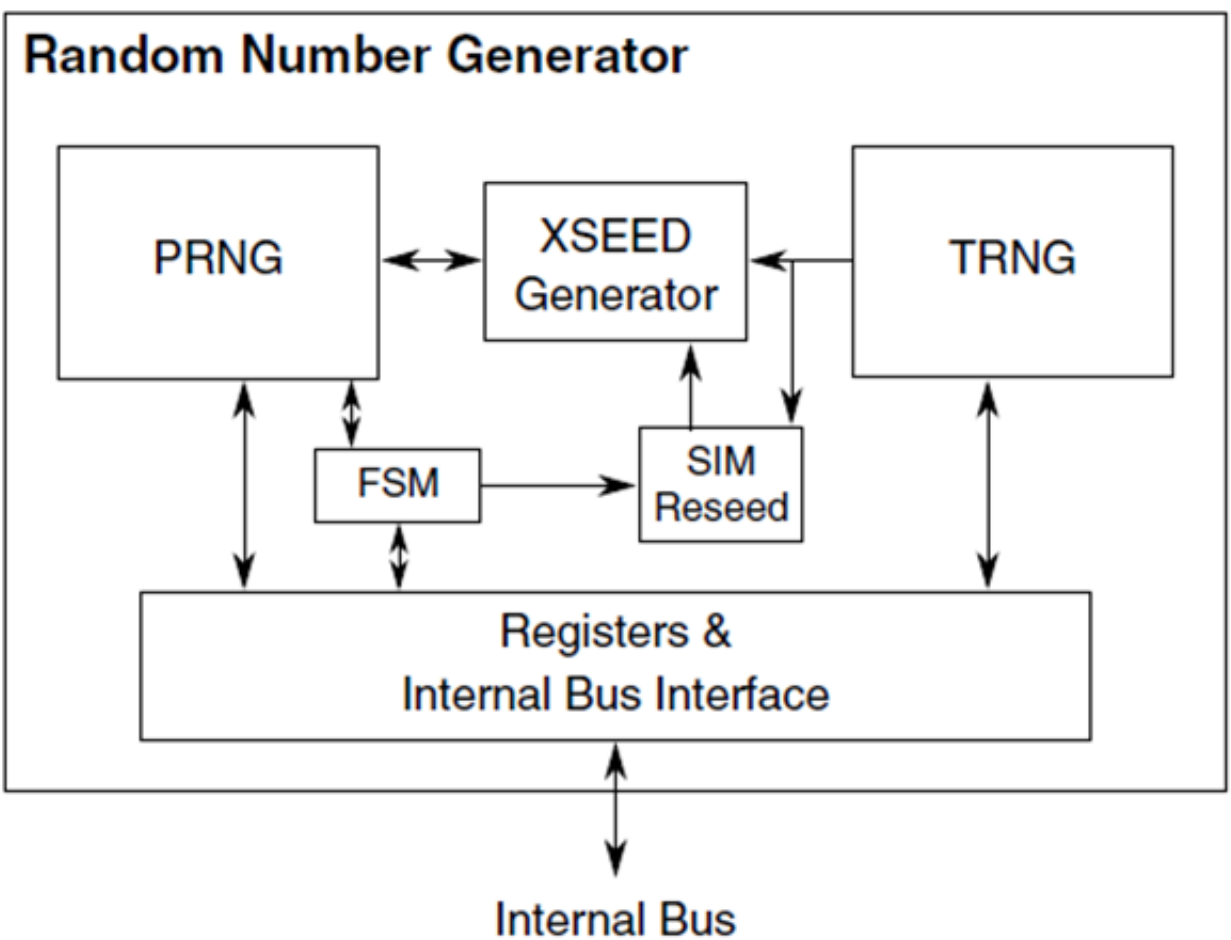


Figure 19. Random number generator

The RNG-A variant is simply a pseudo-random number generator. This module is less bulky and less complex than the RNG-B variant. The RNG-A block diagram is shown in the block diagram below.

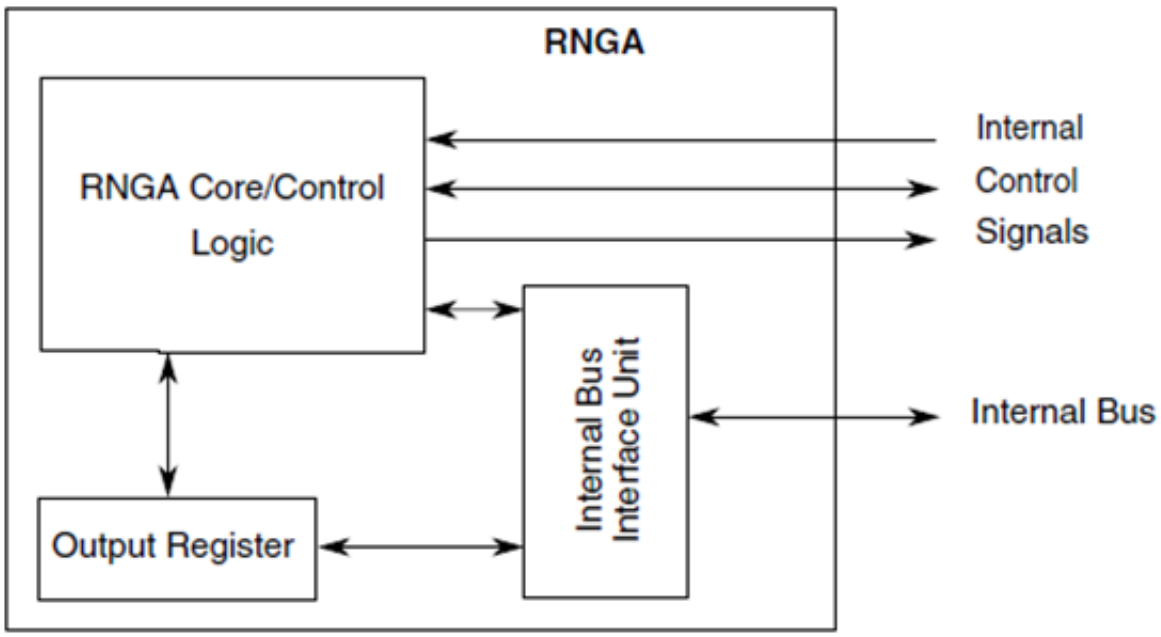


Figure 20. RNG-A block diagram

There is no known cryptographic proof showing that this is a secure method of generating random data. Therefore, it is highly recommended that the random data produced by this module be used as an input seed to a NIST-approved (based on DES or SHA-1) pseudo-random number generator as defined in NIST Fips Pub 186-2 Appendix 3 and NIST Fips Pub SP 800-90.

Though these modules are very different internally, operationally they are very similar. The RNG-A module does not allow for manual seeding. Thus, only the automatic operation algorithms will be compared here. The two different initialization and operation modes are displayed below.

Table 3. Initialization and operation algorithms comparison

RNG-A	RNG-B (Automatic)	RNG-B (Manual)
1. Reset/initialize 2. Write to the RNGA Control Register and set the Interrupt Mask (INTM), High Assurance (HA), and GO bits. 3. Poll the RNGA Status Register for RNGA Output Register level. 4. Read available random data from the RNGA Output Register. 5. Repeat steps 3 and 4 as needed.	1. Reset/initialize. 2. Write to the RNG_CR to set up the RNGB for automatic seeding and the desired functionality. 3. Wait for interrupt to indicate completion of first seed. 4. Poll RNG_SR for FIFO level. 5. Read available random data from output FIFO. 6. Repeat steps 4 and 5 as needed. Automatic seeding occurs when necessary and is transparent to operation.	1. Reset/initialize. 2. Write to the RNG_CR to set up the desired functionality. 3. Write to RNG_CMD register to run self-test or seed generation. 4. Wait for interrupt to indicate completion of the requested operation(s). 5. Repeat steps 3 and 4 if seed generation is not complete. 6. Poll RNG_SR for FIFO level. 7. Read available random data from output FIFO. 8. Repeat steps 6 and 7 as needed, until 2^{20} words have been generated. 9. Write to RNG_CMD to run seed mode. 10. Repeat steps 4 through 9.

3.5.2 Memory map comparison

The RNG-A and RNG-B modules share all of the registers that are present in the RNG-A module with the exception of the entropy register. Therefore, migration to the RNG-A module requires that you remove some code and change the values written to the registers that are shared.

Though they may share registers, the registers are not located in the same memory locations. Therefore, it is important to update your system with the latest Freescale-provided linker files for your system to work correctly.

The changes in the memory map locations are displayed in the figure below. Changes in the reset values for these registers are discussed in the specific register comparisons.

Memory map comparison				
	RNG-B		RNG-A	
	Location	Name	Location	Name
Control Register	400A_0008	RNG_CR	400A_0000	RNG_CR
Status Register	400A_000C	RNG_SR	400A_0004	RNG_SR
Output Register	400A_0014	RNG_OUT	400A_000C	RNG_OR
Entropy Register	N/A	N/A	400A_0008	RNG_ER
Command Register	400A_0004	RNG_CMD	N/A	N/A
Error Status Register	400A_0010	RNG_ESR	N/A	N/A
Version Register	400A_0000	RNG_VER	N/A	N/A

In addition to the changes in location, there have also been changes to the register structures and some reset values. The control register structure of RNG-A and RNG-B is shown below. Note that the register structures are completely different.

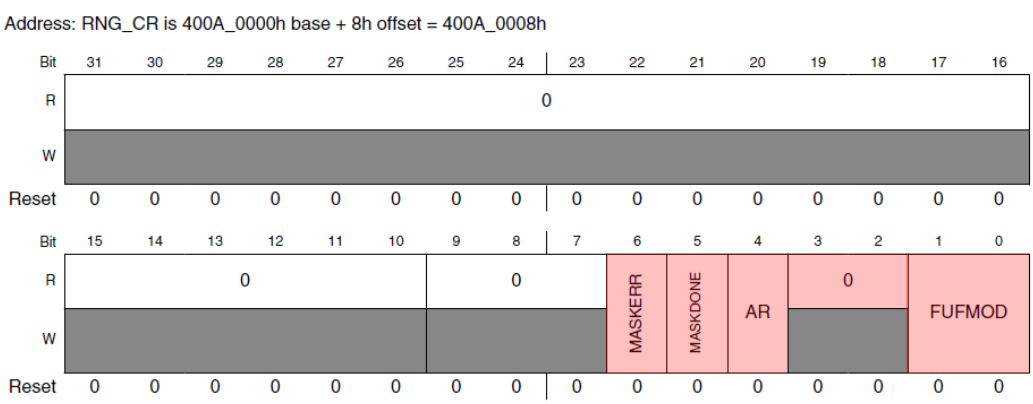


Figure 21. RNG-B module RNG_CR

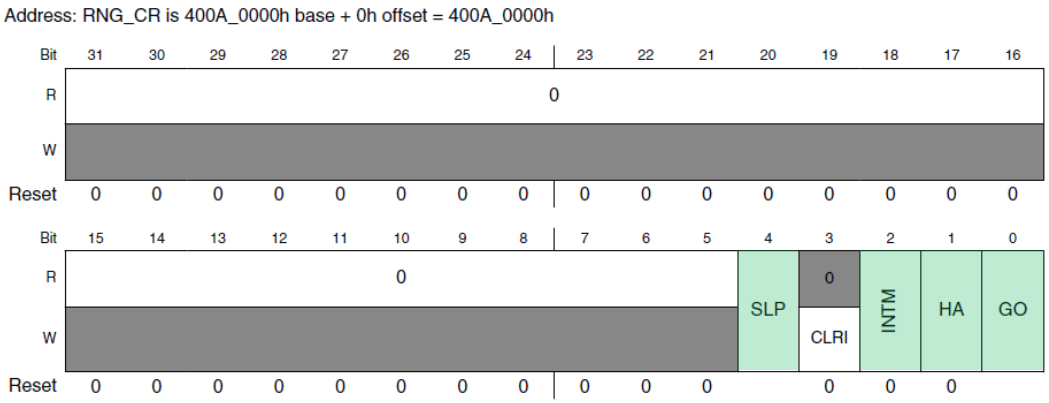


Figure 22. RNG-A module RNG_CR

Deleted bit fields:

- MASKERR
- MASKDONE
- AR
- FUFMOD

Added bit fields:

- SLP: Setting this bit puts the RNG module in Sleep mode; clearing this bit awakens it.
- CLRI: Setting this bit clears the error interrupt flag.
- INTM: Setting this bit enables RNG interrupts; clearing it disables them.
- HA: Enables the security violation bit in the RNG register. Reads of the RNG register while this bit is set are not permitted.
- GO: RNG register is loaded with random data.

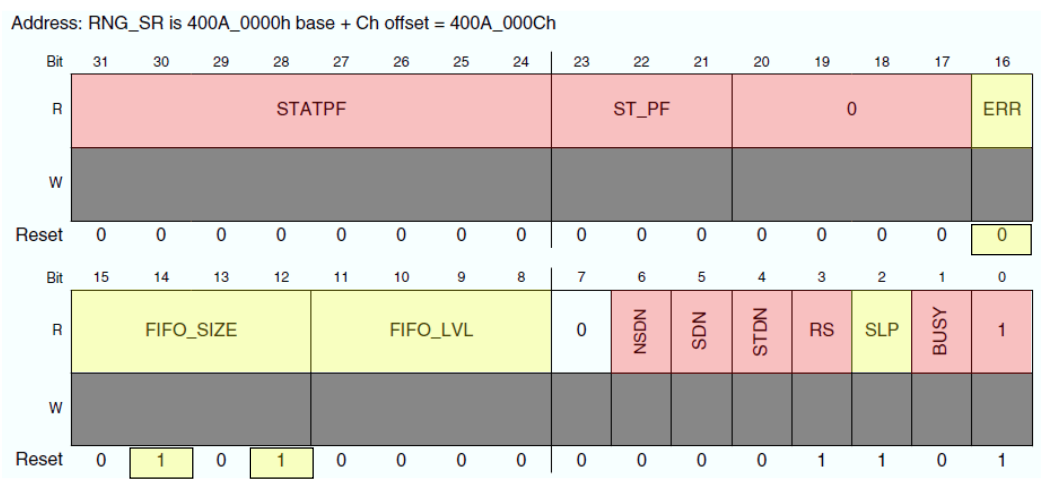


Figure 23. RNG-B module RNG_SR

Address: RNG_ESR is 400A_0000h base + 10h offset = 400A_0010h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0											FUFE	SATE	STE	OSCE	LFE
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 24. RNG-B module RNG_ESR

Address: RNG_SR is 400A_0000h base + 4h offset = 400A_0004h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0								OREG_SIZE							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	OREG_LVL								0			SLP	ERRI	ORU	LRS	SECV
W																
Reset	0	0	0	0	0	0	0	0	0	0	0		0	0	0	0

Figure 25. RNG-A module RNG_SR

Added bit fields:

- SECV: Signals that a security violation has occurred if set. No security violation has occurred if cleared.
- LRS: When set, the last read status bit indicates that the last read was performed while the output register was empty (underflow condition).

Changed bit fields:

- OREG_SIZE: This is an 8-bit integer field that indicates the size of the output register and has been renamed from FIFO_SIZE in the RNG-B variant.
- OREG_LVL: This bit indicates that a random word is available in the output register. Only two values are possible (0b00000001 or 0b00000000). This bit was renamed from FIFO_LVL in the RNG-B variant.
- SLP: Indicates that the RNG module is in Sleep mode when set. This bit was moved from bit 2 (in the RNG-B variant) to bit 4.
- ERRI: When set, this bit indicates that the output register was read while empty. This bit was moved from bit 16 (in the RNG-B variant) to bit 3.
- ORU: When set, this bit indicates that the output register was read while empty since the last read of the status register. This bit was moved from bit 4 of the ESR register (in the RNG-B variant) to bit 2.

Address: RNG_OUT is 400A_0000h base + 14h offset = 400A_0014h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RANDOUT																															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 26. RNG-B module RNG_OUT

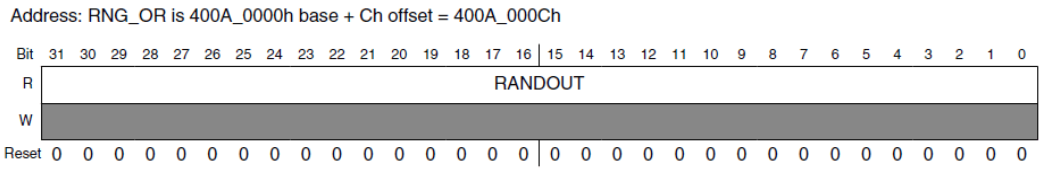


Figure 27. RNG-A module RNG_OR

The output register is simply a 32-bit register that holds the resultant random number once a random number has been generated and, thus, the structure has not been changed. The register has been renamed as shown below.

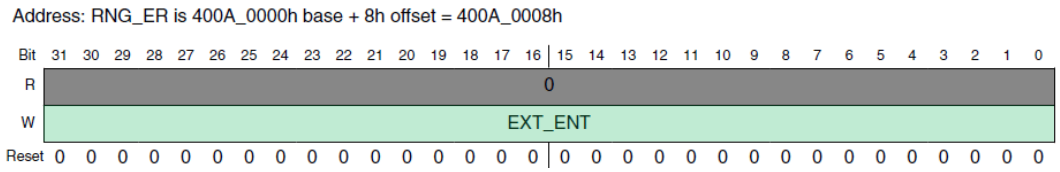


Figure 28. RNG-A module RNG_ER

The RNG-A module allows you to add entropy to the random number generation process. This is done by writing a value to the write-only entropy register. Sources of entropy that may feed the entropy registers are:

- Current time (highest precision possible)
- Mouse and/or keyboard motions
- Other random number generators

3.5.3 Software impact

Freescale Semiconductor will provide customers affected by this migration with new header and linker files, making the migration to the new RNG-A module as seamless as possible. It is important to remember (as previously mentioned in this document) that the RNG-A algorithm does not produce a cryptographically strong random number. If your current software setup relies on a cryptographically strong random number generator for your cryptographic algorithm, you will need NIST-approved support software to generate a cryptographically strong random number to feed into your current software solution. In addition, it is recommended that you use the entropy register of the RNG-A module to add extra entropy to your random number generation process.

There are two general cases for RNG-B to RNG-A migration: either a manually seeded RNG-B algorithm is being implemented or an automatically seeded RNG-B algorithm is being implemented. The manually seeded RNG-B implementation is examined first below.

3.5.4 Impact when using manual seeding algorithm

If you are using manual seeding in your code, you will need to rewrite your code based on the automatic seeding algorithm of RNG-A, as RNG-A provides no manual seeding option. The two algorithms are presented below in a side-by-side comparison highlighting the similarities in the algorithms

RNG-B Manual Algorithm	RNG-A Algorithm
Reset/Initialize	Reset/Initialize
Write to the RNG_CR to setup the desired functionality	Write to the RNGA_CR and set the Interrupt Mask (INTM), High Assurance (HA), and GO bits

Table continues on the next page...

RNG-B Manual Algorithm	RNG-A Algorithm
Write to RNG_CMD register to run self-test or seed generation.	
Wait for interrupt to indicate completion of the requested operation(s).	
Repeat steps 3-4 if seed generation is not complete.	
Poll RNG_SR for FIFO level.	Poll RNG_SR for Output register level.
Read available random data from output FIFO.	Read the available data from the Output Register.
Repeat steps 6 and 7 as needed, until 2^{20} words have been generated.	
Write to RNG_CMD to run seed mode	
Repeat steps 4 through 9.	Repeat steps 3 and 4.

As can be seen from the comparison above, when switching to the RNG-A algorithm, steps 3, 4, 5, and 9 may be deleted from your code project. If you will not be using the Freescale-provided header and linker files, remember that, when writing to the RNGA_CR register, you will need to update the value that is written to the CR register. Also, when polling the SR register, you will to poll bits 8–15 (instead of just 8–11).

3.5.5 Impact when using automatic seeding algorithm

The algorithmic difference between RNG-A and RNG-B are minimal when implementing the automatic seeding functionality. Observe the chart comparison below.

RNG-B Manual Algorithm	RNG-A Manual Algorithm
Reset/initialize	Reset/initialize
Write to the RNG_CR to set up the desired functionality	Write to the RNGA_CR and set the Interrupt Mask (INTM), High Assurance (HA), and GO bits
Wait for interrupt to indicate completion of first seed.	
Poll RNG_SR for FIFO level.	Poll RNG_SR for output register level.
Read available random data from output FIFO.	Read the available data from the output register.
Repeat steps 4 through 9.	Repeat steps 3 and 4.

If you are using the automatic seeding algorithm, you will simply need to remove your code related to the RNG-B seeding interrupt, adjusting the value written to the control register to be the proper value for RNG-A, and adjusting your code to read from the correct output level register. To simplify the output register reads, it is recommended that you insert a define statement converting the output register name of RNG-B to the output register name of the RNG-A output register.

```
(#define RNG_OUT    RNG_OR).
```

3.5.6 Hardware impact

None

3.6 SSI to SAI

There have been significant change from the SSI module on Kinetis Rev. 1.x to the SAI module on Kinetis Rev. 2.x, including differences in the memory map as well as the features supported.

3.6.1 Memory map comparison

Table 7 provides a memory map comparison of the SSI and SAI modules. Since the memory map and the registers' intended functions have changed significantly in some cases, the table uses the following conventions:

1. Registers that implement similar functions are listed in the same row; for example, I2S0_TX0 corresponds to I2S0_TDR0.
2. If there is an SSI module register whose function needs to be implemented with more than one register, that one register map is merged so that it corresponds to all those related registers in the SAI module; for example, I2S0_TCR's function needs to be implemented with I2S0_TCR2, I2S0_TCR3, and I2S0_TCR4.
3. Some of the registers in either the SSI or SAI module do not have a related register in the other module; for example, there is no hardware acceleration support for AC97 on SAI and no related AC97 registers on SAI. In these cases, "N/A" in the associated row indicates that the register is not available.
4. For the SSI module, master clock generation needs to configure two registers in the SIM module, while SAI has pulled related registers into its own memory space.

Table 7. Memory map comparison

	Kinetis Rev. 1.x (SSI)			Kinetis Rev. 2.x (SAI)	
I2S transmit data register 0	4002_F000	I2S0_TX0	SAI transmit data register 0	4002_F020	I2S0_TDR0
I2S transmit data register 1	4002_F004	I2S0_TX1	SAI transmit data register 1	4002_F024	I2S0_TDR1
I2S receive data register 0	4002_F008	I2S0_RX0	SAI receive data register 0	4002_F0A0	I2S0_RDR0
I2S receive data register 1	4002_F00C	I2S0_RX1	SAI receive data register 1	4002_F0A4	I2S0_RDR1
I2S control register	4002_F010	I2S0_CR	SAI transmit control register	4002_F000	I2S0_TCSR
			SAI receive control register	4002_F080	I2S0_RCSR
			SAI transmit configuration 2 register	4002_F008	I2S0_TCR2
			SAI receive configuration 2 register	4002_F088	I2S0_RCR2
			SAI transmit configuration 3 register	4002_F00C	I2S0_TCR3
			SAI receive configuration 3 register	4002_F08C	I2S0_RCR3

Table continues on the next page...

Table 7. Memory map comparison (continued)

	Kinetis Rev. 1.x (SSI)			Kinetis Rev. 2.x (SAI)	
I2S interrupt status register	4002_F014	I2S0_ISR	SAI transmit control register	4002_F000	I2S0_TCSR
			SAI receive control register	4002_F080	I2S0_RCSR
I2S interrupt enable register	4002_F018	I2S0_IER	SAI transmit control register	4002_F000	I2S0_TCSR
			SAI receive control register	4002_F080	I2S0_RCSR
I2S transmit configuration register	4002_F01C	I2S0_TCR	SAI transmit configuration 2 register	4002_F008	I2S0_TCR2
			SAI transmit configuration 3 register	4002_F00C	I2S0_TCR3
			SAI transmit configuration 4 register	4002_F010	I2S0_TCR4
I2S receive configuration register	4002_F020	I2S0_RCR	SAI receive configuration 2 register	4002_F088	I2S0_RCR2
			SAI receive configuration 3 register	4002_F08C	I2S0_RCR3
			SAI receive configuration 4 register	4002_F090	I2S0_RCR4
I2S transmit clock control register	4002_F024	I2S0_TCCR	SAI transmit configuration 2 register	4002_F008	I2S0_TCR2
			SAI transmit configuration 4 register	4002_F010	I2S0_TCR4
			SAI transmit configuration 5 register	4002_F014	I2S0_TCR5
I2S receive clock control register	4002_F028	I2S0_RCCR	SAI receive configuration 2 register	4002_F088	I2S0_RCR2
			SAI receive configuration 4 register	4002_F090	I2S0_RCR4
			SAI receive configuration 5 register	4002_F094	I2S0_RCR5

Table continues on the next page...

Table 7. Memory map comparison (continued)

	Kinetis Rev. 1.x (SSI)			Kinetis Rev. 2.x (SAI)	
I2S FIFO control/ status register	4002_F02C	I2S0_FCSR	SAI transmit configuration 1 register	4002_F004	I2S0_TCR1
			SAI receive configuration 1 register	4002_F084	I2S0_RCR1
			SAI transmit FIFO 0 register	4002_F040	I2S0_TFR0
			SAI transmit FIFO 1 register	4002_F044	I2S0_TFR1
			SAI receive FIFO 0 register	4002_F0C0	I2S0_RFR0
			SAI receive FIFO 1 register	4002_F0C4	I2S0_RFR1
I2S0 transmit time slot mask register	4002_F048	I2S0_TMSK	SAI transmit mask register	4002_F060	I2S0_TMR
I2S0 receive time slot mask register	4002_F04C	I2S0_RMSK	SAI receive mask register	4002_F0E0	I2S0_RMR
I2S AC97 command data register	4002_F040	I2S0_ACDAT	N/A	N/A	N/A
I2S AC97 tag register	4002_F044	I2S0_ATAG	N/A	N/A	N/A
I2S AC97 Channel status register	4002_F050	I2S0_ACCST	N/A	N/A	N/A
I2S AC97 channel enable register	4002_F054	I2S0_ACCEN	N/A	N/A	N/A
I2S AC97 channel disable register	4002_F058	I2S0_ACCDIS	N/A	N/A	N/A
System option register 2	4004_8004	SIM_SOPT2	SAI MCLK control register	4002_F100	I2S0_MCR
System clock divider register 2	4004_8048	SIM_CLKDIV2	SAI MCLK divider register	4002_F014	I2S0_MDR

3.6.2 Features

Even though the register map and bit definitions of the SAI module are quite different from the SSI module, the main features are still the same. SAI supports full duplex synchronous serial interfaces with frame sync such as I2S, TDM, AC97, codec, and DSP interfaces. However, in the movement from the SSI to the SAI module, some features have been added and some have been removed:

Following is a list of what has been removed on the SAI module:

- No option to disable transmit and receive frame sync separately. Once TE or RE enabled, frame sync generates, it can only be disabled by clearing TE or RE.
- No option to output oversampling clock on RX bit clock.

- No separate bits to control whether SAI operates under I2S mode or network mode; it all depends on how you configure the frame sync size.
- No longer supports gated clock mode.
- No hardware support for AC97, though AC97 can still be supported with software if configured for 13 or more words per frame.
- No support for separate start-of-frame flag and start-of-last-unmasked-word-in-frame-flag. These have been replaced with single flag to represent the start of a word in a frame.
- No option to disable TX or RX FIFOs. They are always enabled.
- No support for separate left/right FIFOs. Instead there are two separate pins for TX and RX, each associated with its own FIFO.
- No option to select MSB aligned or LSB aligned
- No bits to represent FIFO entry counts. Instead, a write and read FIFO pointer is used to represent current FIFO entry and can be used to tell whether the FIFO is full or empty.

Below is a list of what is new on the SAI module

- Support for Stop mode operation
- Support for Debug mode operation
- Support for 32-bit transfers
- Added option to enable and disable bit clock separately
- Added frame sync error flag
- Added TX FIFO empty or RX FIFO full flag to allow trigger interrupt and DMA, apart from the TX and RX FIFO watermark trigger
- One additional TX and RX data channel
- SAI master clock select and divide registers relocated to SAI's own memory space. (For Kinetis Rev. 1.x, these were located in the SIM module.)

3.6.3 Control register

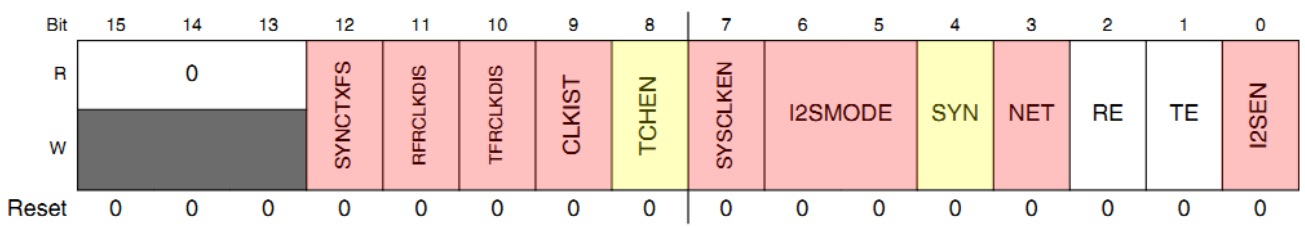


Figure 29. I2S0_CR–Rev. 1.x

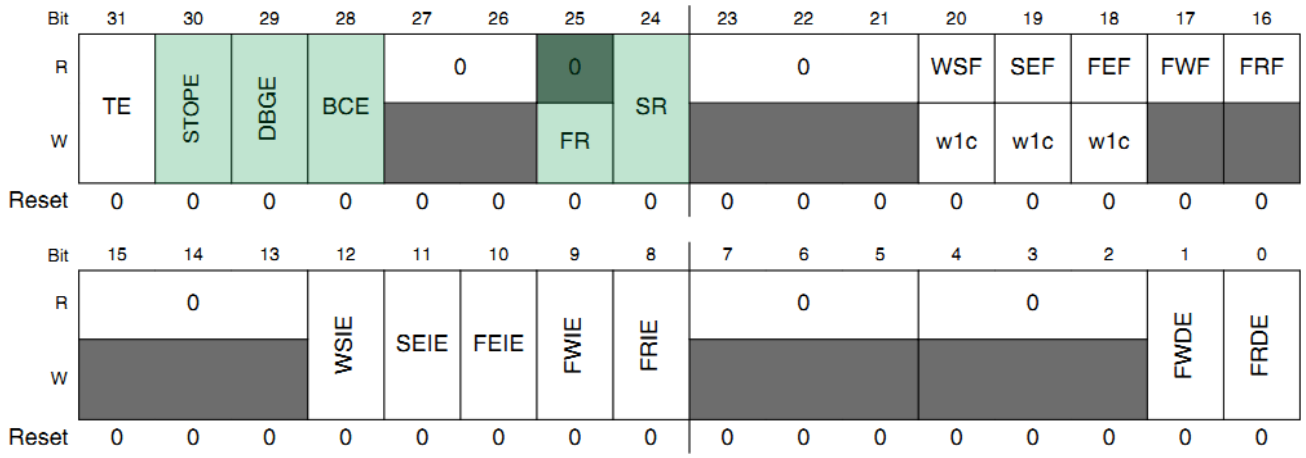


Figure 30. I2S0_TCSR–Rev. 2.x

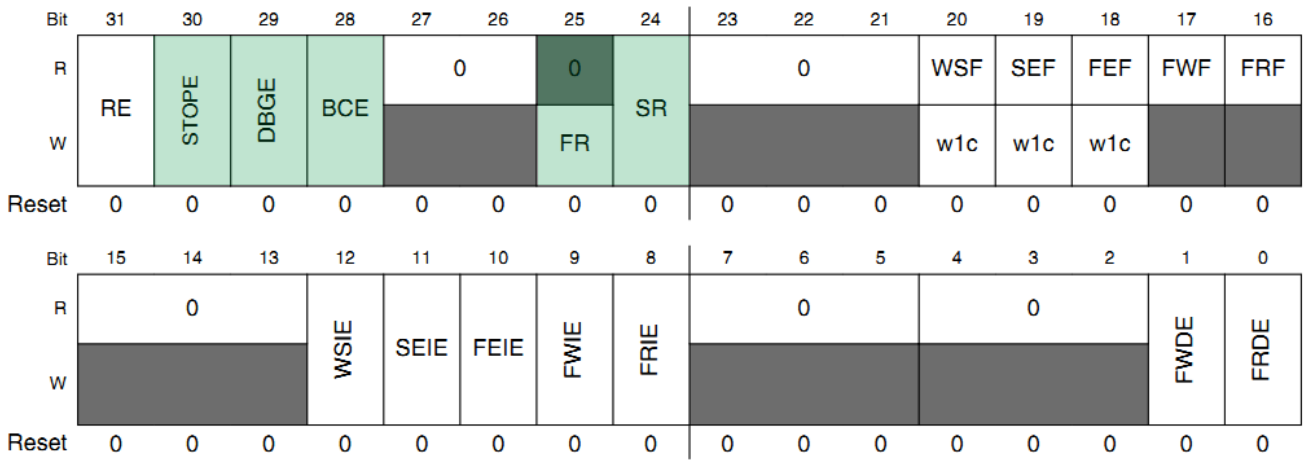


Figure 31. I2S0_RCSR–Rev. 2.x

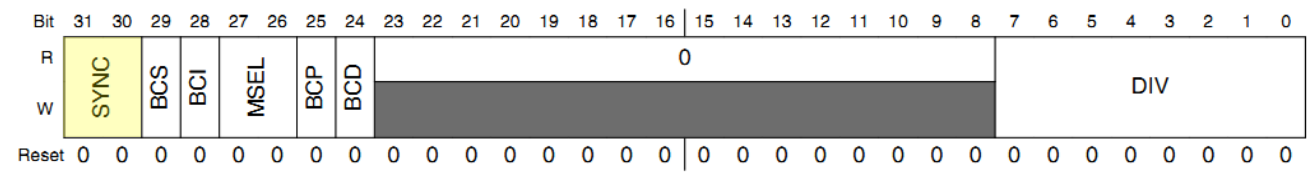


Figure 32. I2S0_TCR2 and I2S0_RCR2–Rev. 2.x

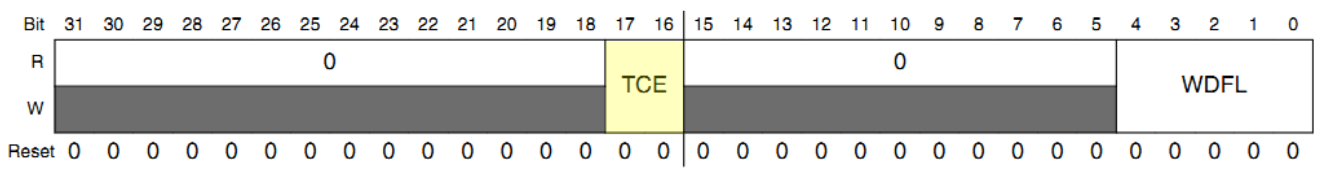


Figure 33. I2S0_TCR3–Rev. 2.x

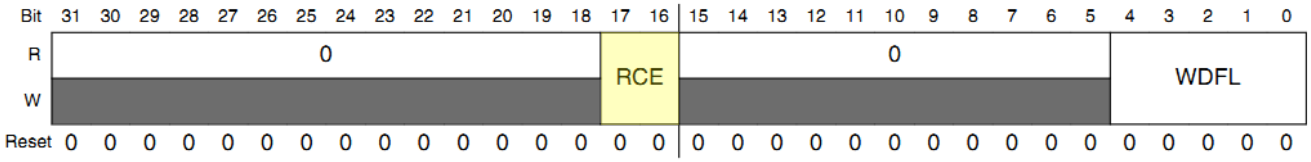


Figure 34. I2S0_RCR3–Rev. 2.x

New bits/fields added:

- **STOPE:** Stop Enable
- **DBGE:** Debug Enable
- **FR:** FIFO Reset
- **SR:**Software Reset

Changed bits/fields name:

- TCHEN → TCE and RCE
- SYN → SYNC

Removed bits/fields:

- **SYNCTXFS:** CR[TE] latch with FS occurrence
- **RFRCLKDIS:** Receive Frame Sync Disable
- **TFRCLKDIS:** Transmit Frame Sync Disable
- **CLKIST:** Clock Idle State during I2S Gated Clock Mode
- **SYSCLKEN:** Oversampling Clock Enable
- **I2SMODE:** I2S Mode Select
- **NET:** Network Mode I2SEN: I2S enable

3.6.4 Interrupt and status enable register

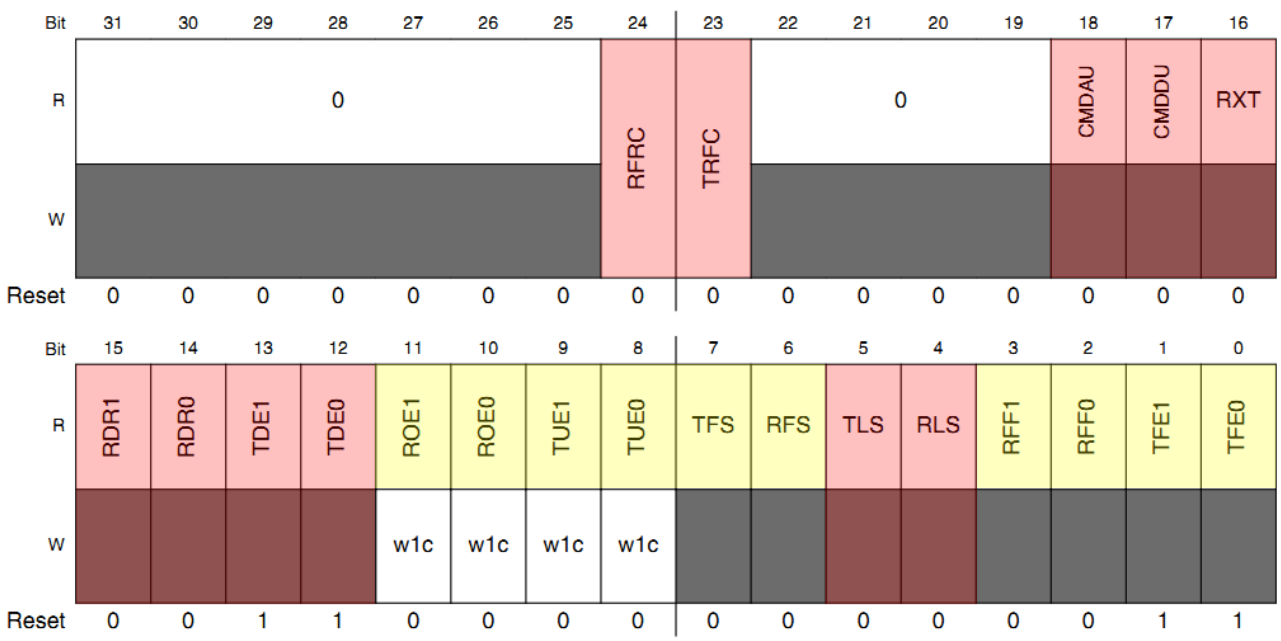


Figure 35. I2S0_ISR—Rev. 1.x

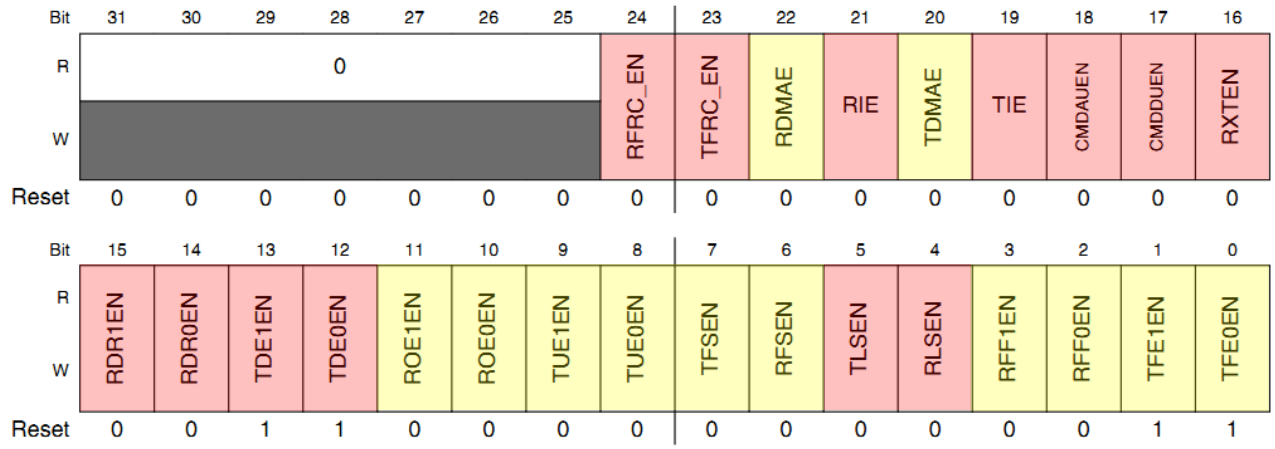


Figure 36. I2S0_IER—Rev. 1.x

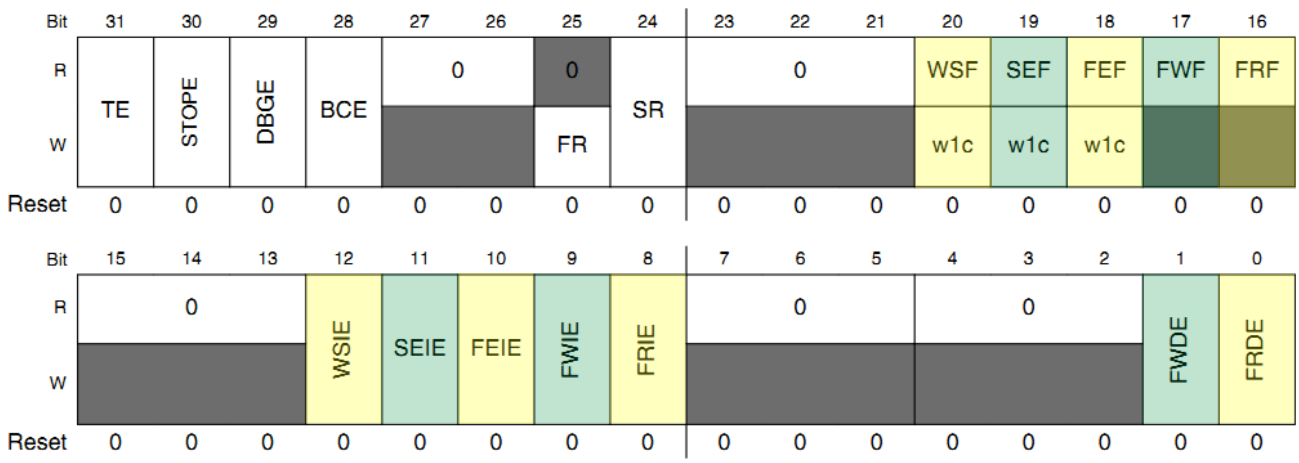


Figure 37. I2S0_TCSR—Rev. 2.x

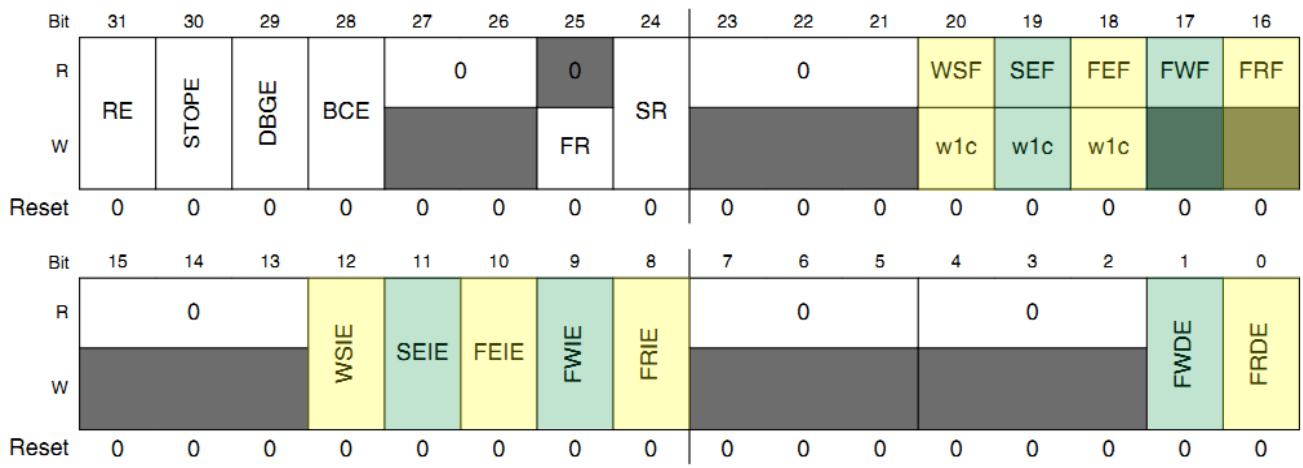


Figure 38. I2S0_RCSR—Rev. 2.x

New bits/fields added:

- **SEF:** Sync Error Flag
- **FWF:** FIFO Warning Flag
- **SEIE:** Sync Error Interrupt Enable

- **FWIE:** FIFO Warning Interrupt Enable
- **FWDE:** FIFO Warning DMA Enable

Changed bits/fields name:

- ROE1 & ROE0 → RCSR[FEF] (Receive FIFO Error, overrun)
- TUE1 & TUE0 → TCSR[FEF] (Transmit FIFO Error, underrun)
- TFS → TCSR[WSF] (Transmit Frame Sync)
- RFS → RCSR[WSF] (Receive Frame Sync)
- RFF1 & RFF0 → RCSR[FRF] (Receive FIFO Request, received data greater than watermark)
- TFE1 & TFE0 → TCSR[FRF] (Transmit FIFO Request, transmit data less than watermark)
- RDMAE → RCSR[FRDE] (Receive FIFO Request DMA Enable)
- TDMAE → TCSR[FRDE] (Transmit FIFO Request DMA Enable)
- ROE1EN & ROE0EN → RCSR[FEIE]
- TUE1EN & TUE0EN → TCSR[FEIE]
- TFSEN → TCSR[WSIE]
- RFSEN → RCSR[WSIE]
- RFF1EN and RFF0EN → RCSR[FRIE]
- TFE1EN and TFE0EN → TCSR[FRIE]

Removed bits/fields:

- **TFRC and RFRC:** Transmit and Receive Frame Complete
- **TLS and RLS:** Transmit and Receive Last Time Slot
- **RDR1 and RDR0:** Receive Data Ready
- **TDE1 and TDE0:** Transmit Data Empty
- **CMDAU:** Command Address Register Updated
- **CMDDU:** Command Data Register Updated
- **RXT:** Receive Tag Updated

Associated interrupt enable bits:

- TFRCEN and RFRCEN,
- TLSEN and RLSEN,
- RDR1EN and RDR0EN,
- TDE1EN, and
- TDE0E, CMDAUEN, CMDDUEN, RXTEN, RIE, TIE

3.6.5 Transmit and receive configuration register

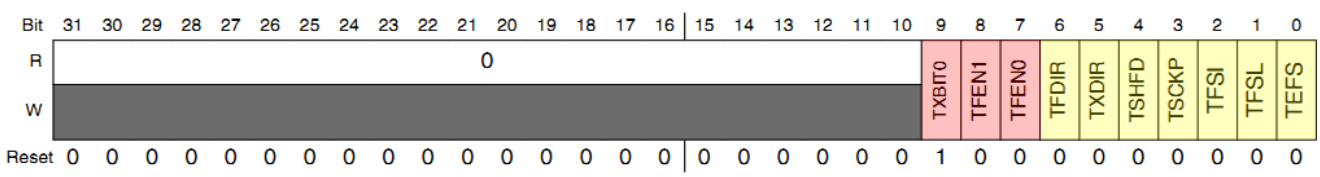


Figure 39. I2S0_TCR—Rev. 1.x

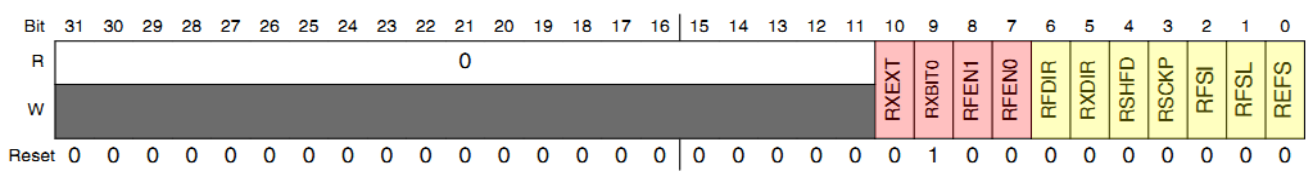


Figure 40. I2S0_RCR—Rev. 1.x

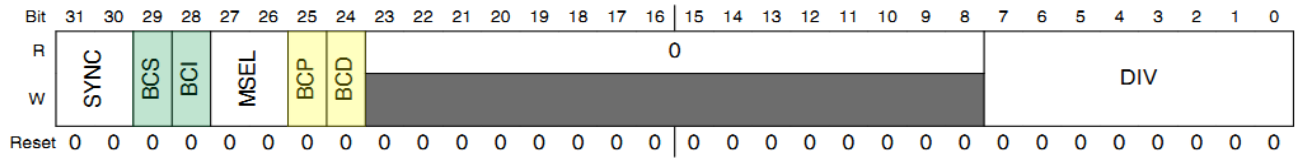


Figure 41. I2S0_TCR2 and I2S0_RCR2—Rev. 2.x

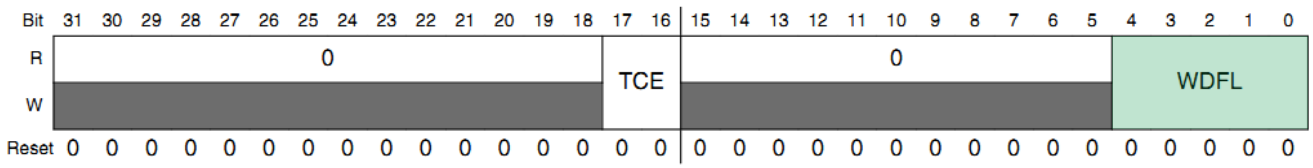


Figure 42. I2S0_TCR3—Rev. 2.x

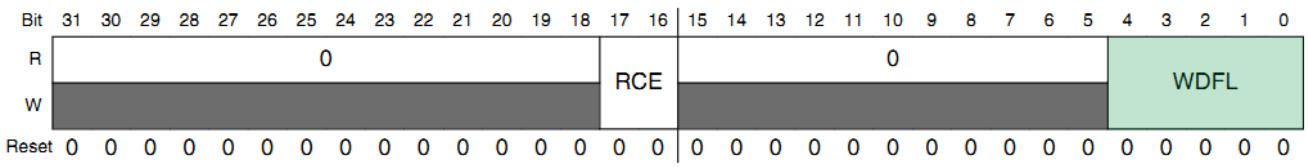


Figure 43. I2S0_RCR3—Rev. 2.x

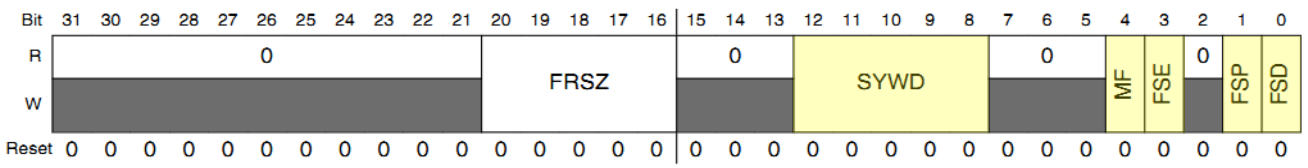


Figure 44. I2S0_TCR4 and I2S0_RCR4—Rev. 2.x

New bits/fields added:

- **BCS:** Bit Clock Swap
- **BCI:** Bit Clock Input
- **WDFL:** Configures which word start of word flag is set

Changed bits/fields name:

- TSCKP and RSCKP → BCP (Bit Clock Polarity)
- TXDIR and RXDIR → BCD (Bit Clock Direction)
- TFSL and RFSL → SYWD (Frame sync length)
- TSHFD and RSHFD → MF (MSB or LSB transmit first)
- TEFS and REFS → FSE (Frame Sync Early)
- TFSI and RFSI → FSP (Frame Sync Polarity)
- TFDIR and RFDIR → FSD (Frame Sync Direction)

Removed bits/fields:

- **TXBIT0 and RXBIT0:** MSB aligned or LSB aligned
- **TFEN0 and TFEN1:** Transmit FIFO Enable
- **RXEXT:** Receive Sign Extension
- **RFEN0 and RFEN1:** Receive FIFO Enable

3.6.6 Transmit and receive clock configuration register

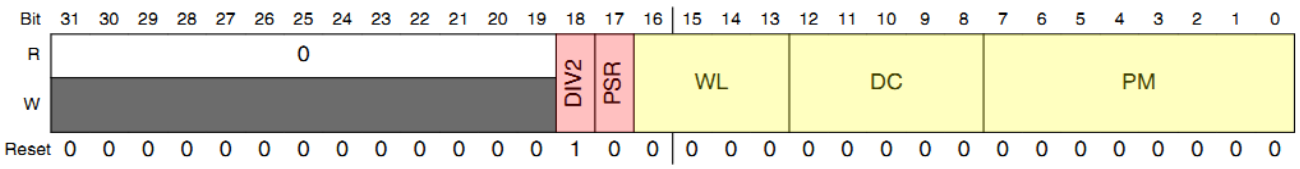


Figure 45. I2S0_TCCR and I2S0_RCCR—Rev. 1.x

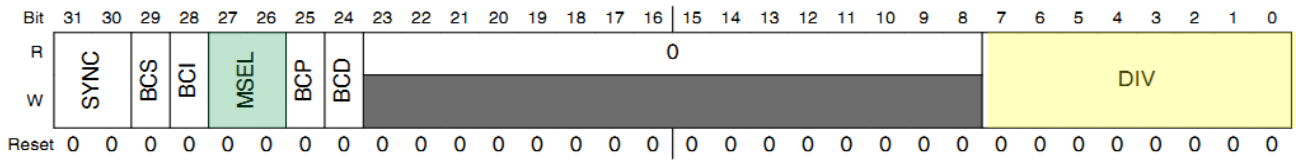


Figure 46. I2S0_TCR2 and I2S0_RCR2—Rev. 2.x

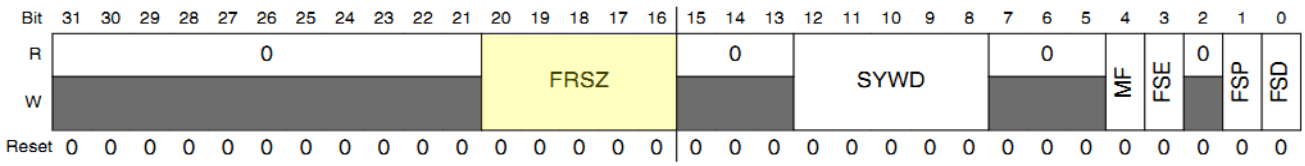


Figure 47. I2S0_TCR4 and I2S0_RCR4—Rev. 2.x

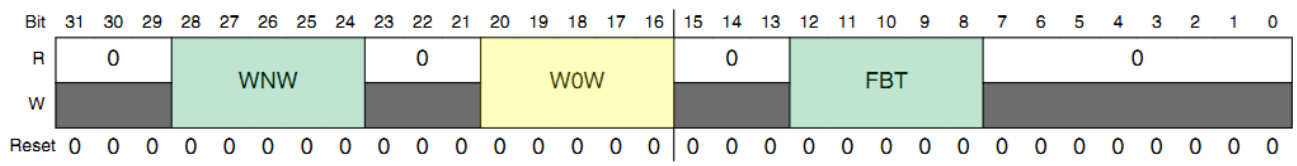


Figure 48. I2S0_TCR5 and I2S0_RCR5—Rev. 2.x

New bits/fields added:

- MSEL: Master Clock Select before it is divided down to bit clock
- WNW: Configures number of bits of each word, except the first word in a frame
- FBT: Configures the bit index of the first bit transmitted in each word in a frame

Changed bits/fields name:

- DC → FRSZ (number of words in each frame)
- WL → W0W (number of bits in each word)
- PM → DIV (bit clock prescaler)

Removed bits/fields:

- DIV2: Whether or not to divide by 2
- PSR: Prescaler divide by 8

3.6.7 FIFO control and status register

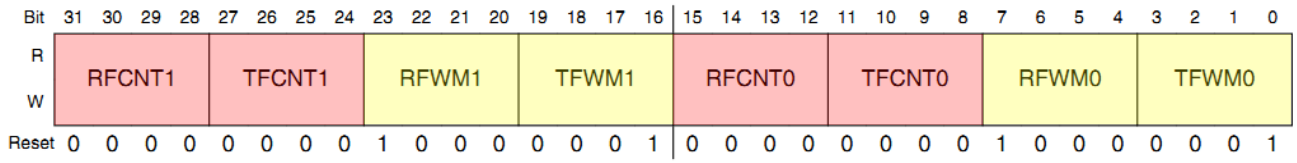


Figure 49. I2S0_FCSR—Rev. 1.x

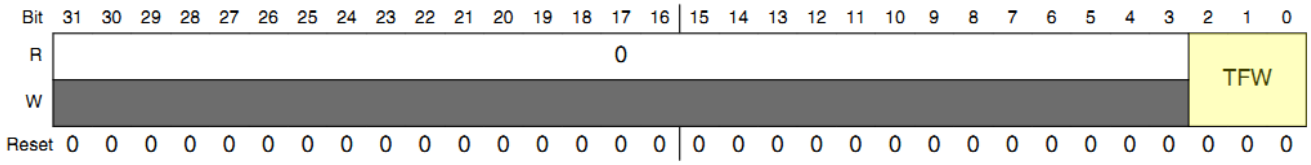


Figure 50. I2S0_TCR1—Rev. 2.x

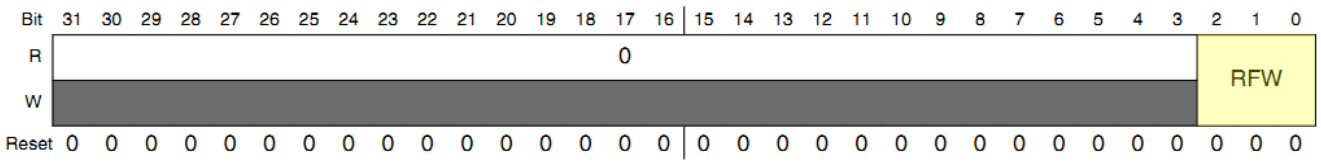


Figure 51. I2S0_RCR1—Rev. 2.x

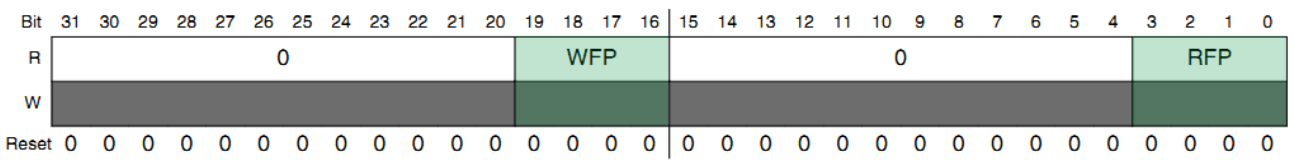


Figure 52. I2S0_TFR0, I2S0_TFR1, I2S0_RFR0, and I2S0_RFR1—Rev. 2.x

New bits/fields added:

- **WFP:** Write FIFO Pointer
- **RFP:** Read FIFO Pointer

Changed bits/fields name:

- RFWM0 and RFWM1 → RFW (Receive FIFO Watermark)
- TFWM0 and TFWM1 → TFW (Transmit FIFO Watermark)

Removed bits/fields:

- RFCNT0 and RFCNT1: Receive FIFO Counter
- TFCNT0 and TFCNT1: Transmit FIFO Counter

3.6.8 Master clock generation register

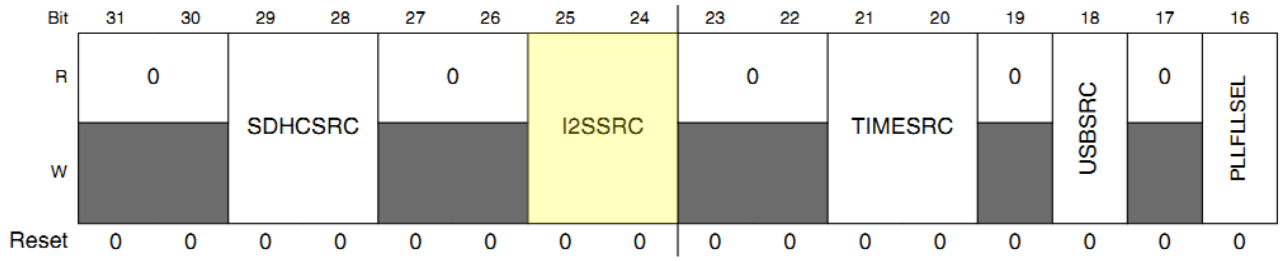


Figure 53. SIM_SOPT2—Rev. 1.x

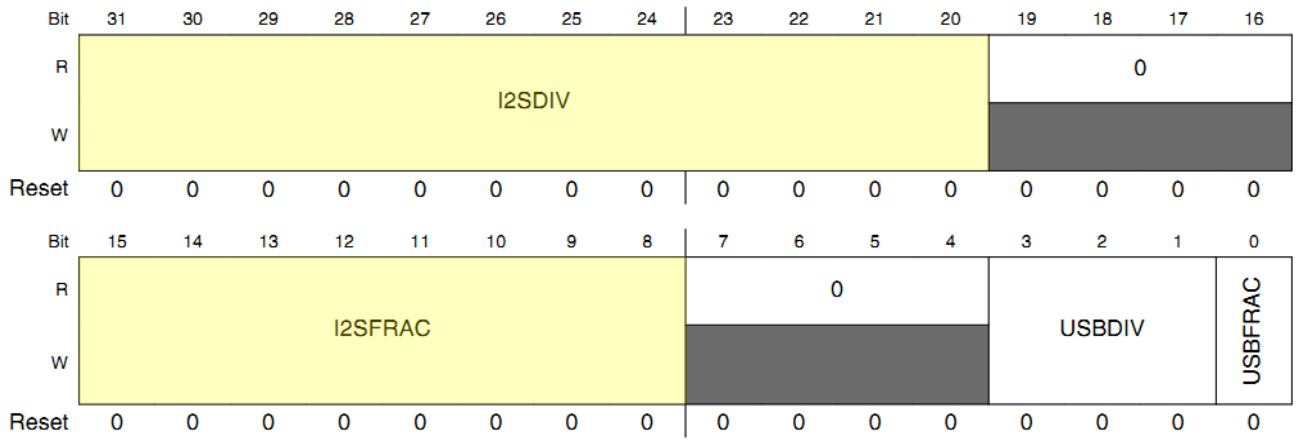


Figure 54. SIM_CLKDIV2—Rev. 1.x

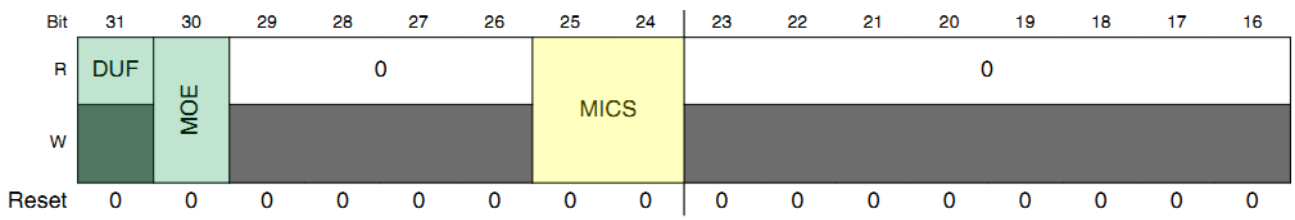


Figure 55. I2S0_MCR—Rev. 2.x

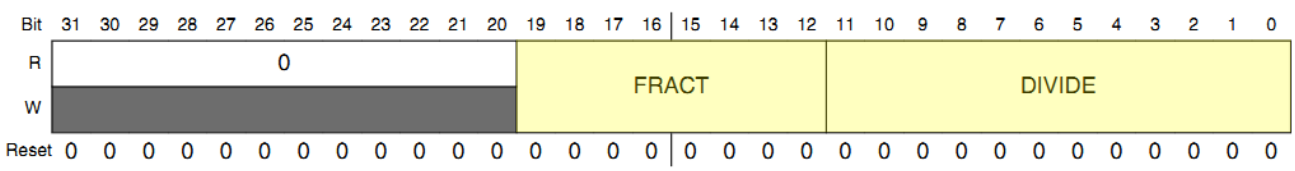


Figure 56. I2S0_MDR—Rev. 2.x

New bits/fields added:

- DUF: Divide Update Flag
- MOE: Master Clock Output Enable

Changed bits/fields name:

- I2SSRC → MICS (Master Clock Source Select)
- I2SFRAC → FRACT (Clock Divider Fraction)

3.6.9 Software impact

There is not much change in software configuration for the SAI module. Bit clock and frame sync generation, their direction and polarity selection, frame size, and word size setting are the same as for the SSI module. Only the bit naming is different.

With the option to trigger DMA with a TX FIFO empty or RX FIFO full condition, it is easier to program the DMA controller for automatic data transfer by simply programming the total transfer size to be the same as the FIFO depth for the TX and RX FIFO. If instead the FIFO watermark condition is used to trigger DMA, take care to program the DMA controller with a total transfer size less than (FIFO depth – FIFO watermark); otherwise, either TX FIFO is pushed with more data than it can hold, or the RX FIFO is pulled with more data that has been received.

Also note that there is no longer any FIFO count information; instead we have write and read FIFO pointer, so the actual FIFO entry needs to be calculated in your code. Also you need to include logic to determine whether the FIFO is empty or full, and to stop writing data when the FIFO is full or reading data when the FIFO is empty.

3.6.10 Hardware impact

There is not much hardware change from the SSI to the SAI. There is one more data channel for TX and RX, so there are I2S0_TXD0 and I2S0_TXD1, as well as I2S0_RXD0 and I2S0_RXD1. It makes no difference whether you use channel 0 or 1 for transmit or receive data. Since each channel has its own FIFO, each channel can send or receive same or different data and do this at the same time they share the same frame sync and bit clock.

3.7 TSI Version 1 to TSI Version 2

This section addresses the specific differences between the TSI module for Kinetis Rev. 1.x and Kinetis Rev. 2.x. The TSI module has been simplified in the new version to simplify configuration. The current ranges for the oscillator remain the same, but the number of possible configuration values has been reduced by half. Delta voltage configuration was removed; now, a constant delta voltage is used. Threshold registers used to generate an interrupt when capacitance increases have been removed for each channel; only one register was kept for the low-power wake-up channel, exclusively. Linked to this change, the status register for all these thresholds was also removed; now the threshold is only used to wake up. The changes in software needed to adjust for these differences are detailed in the sections below.

3.7.1 Memory map comparison

	Kinetis Rev. 1.x (TSI 1.0)		Kinetis Rev. 2.x (TSI 2.0)	
	Location	**Name**	**Location**	**Name**
General Control and Status Register	4004_5000	TSI0_GENCS	4004_5000	TSI0_GENCS
SCAN control register	4004_5004	TSI0_SCANC	4004_5004	TSI0_SCANC
Pin enable register	4004_5008	TSI0_PEN	4004_5008	TSI0_PEN
Status Register	4004_500C	TSI0_STATUS	N/A	N/A
Counter Registers	4004_5100–4004_511C	TSI0_CNTRn	4004_5100–4004_511C	TSI0_CNTRn
Channel threshold register(s)	4004_5120–4004_515C	TSI0_THRESHLDn	4004_5120	TSI0_THRESHOLD
Wake-up counter	N/A	N/A	4004_500C	TSI_WUCNTR

- **GENCS**: No change, but the out-of-range setting is now only operational in low-power modes.
- **SCANC: CAPTRM** removed (internal cap is now fixed to 1 pF), **REFCHRG** and **EXTCHRG** have been scaled down from 5 bits to 4 bits. Change involves reducing the amount of possible current values from 32 to 16. Current range is now from 2 µA to 32 µA in increments of 2. **DELVOL** eliminated, delta voltage value is now fixed. AMCLKS clock sources changed: bus clock reference was changed for **LPOCLK**. **AMCLKSDIV** was also removed because it was only practical with bus clock as reference. Instead of dividing down the clock source, now a slower, lower power source can be used. This reduces overall power consumption of the module. If a slow source is needed, use **LPOCLOCK** and adjust with **AMPSC**. If a faster source is needed, use **MCGIRCLK** or **OSCERCLK**.

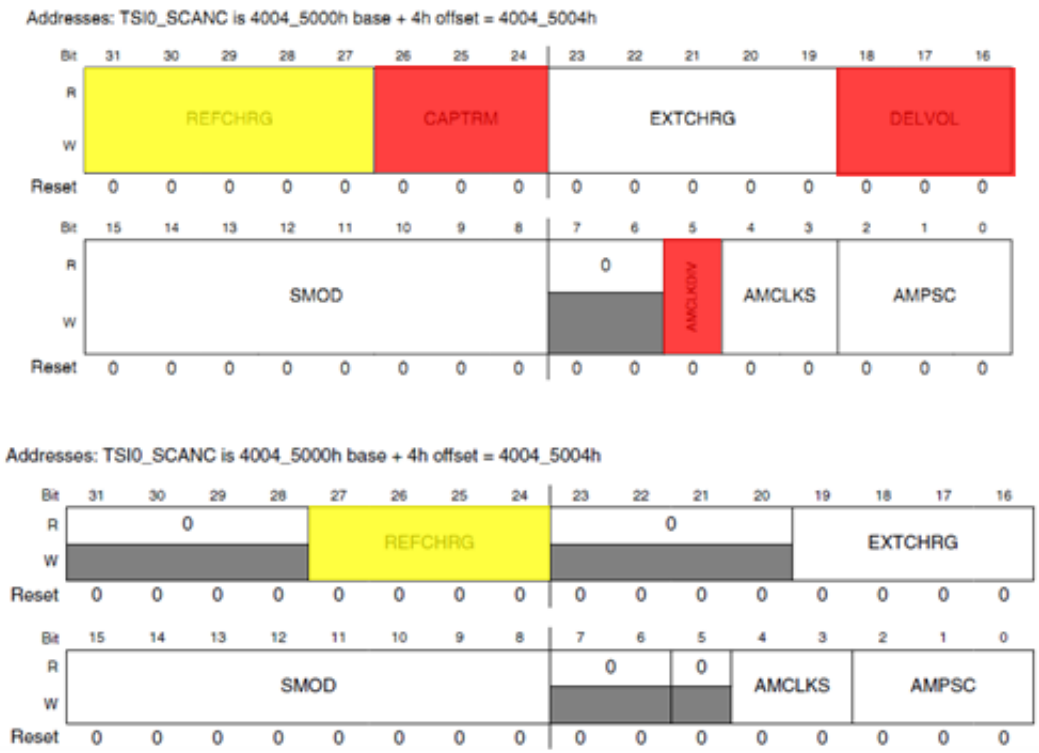
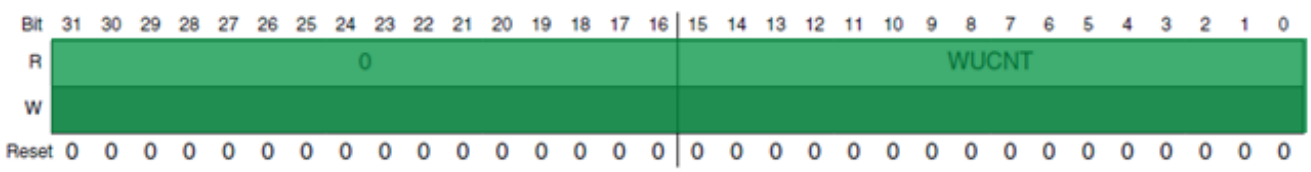


Figure 57. SCANC registers (old and new versions)

- **TSI_PEN:** No change
- **TSI_WUCNTR:** New register, stores the value of the counter for the low power electrode upon wakeup.



TSIx_WUCNTR field descriptions

- **TSI_STATUS:** Removed

Figure 58. TSI_STATUS (completely removed)

- **TSI_CNTR:** No change
- **TSI_THRESHLDn:** Used to be one threshold register for each electrode; now there is a single threshold register for the selected wakeup electrode because the out-of-range function now only operates in low-power modes.

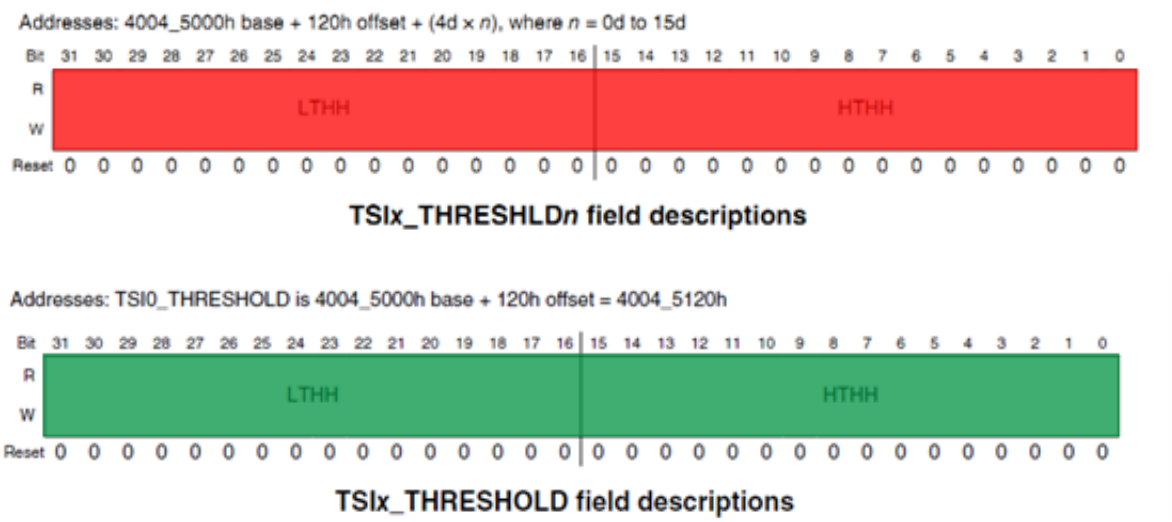


Figure 59. Single TSI_THRESHOLD for wakeup instead of separate registers for each channel

3.7.2 Software impact

- No delta voltages register because there is no difference between internal delta voltage and the external, so the frequency relationship is not affected. The highest voltage that can be output has been selected as a fixed voltage to increase EMC tolerance. As the change in delta voltages does not affect the relationship between oscillators and thus, the counters, this change does not affect software other than the need to remove the code that configured this register.

- Less current for external and internal oscillators. Touch sensing does not require the level of granularity that was in the first TSI version, so the number of values has been scaled down to 16. It is important to note that this change caused the EXTCHRG and REFCHRG registers to go from 5 to 4 bits, meaning that it is necessary to scale down the value. Make sure to take care of this modification. If the value used previously is now not valid, choose either the next value up or down depending on convenience.
- The change of bus clock to LPOCLK for timing reference in AMCLKS means that the prescaler (AMPSC) and modulo (SMOD) need to be adjusted to the LPOCLK, which is a 1 kHz reference. If the other two clock sources were used (MCGIRCLK or OCSERCLK), no change is needed.
- Only one threshold register instead of one for each counter. Consider that the wakeup source can wake up from EOSF (end of scan) or out of range. Because out of range is only available in the low leakage modes (LLS and VLLSx), it is recommended that any of these be used to wake up with TSI. If another mode is needed (like Stop, Low-power Stop, Wait, and so on) and it is needed to wake-up with TSI, then it is important to have a periodic wakeup to check the TSI status. The periodic wakeup can be an external source like RTC or MTIM, or can be the TSI module by configuring the scan time to be very slow (for example, every two or three seconds). This will cause reaction time to be slow, but also lower power.
- Remove references to TSI_STATUS, as it is no longer available.
- Remove references to the numbered TSI_THRESHLDn registers; remember now only one TSI_THRESHOLD register, with the OURGF flag and interrupt, is used for wakeup.
- If the first counter value upon wakeup is needed (from the configured wake-up electrode), use TSI_WUCNTR to read this value. If not, wait for next scan and read directly from the counter registers.

3.7.3 Hardware impact

There is no hardware impact. The signaling and measurement algorithm are the same.

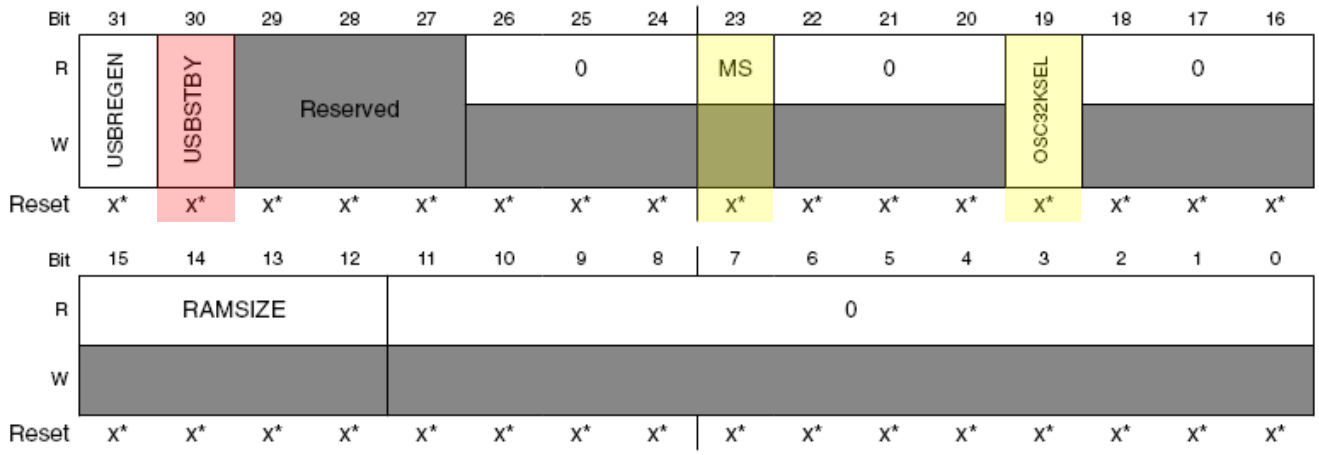
4 Enhanced Modules

4.1 System Integration Module (SIM)

4.1.1 Impacted register

The SIM module consists of new/removed registers along with bit fields that have been added, removed, and changed.

The affected registers are summarized below.



* Notes:

- x = Undefined at reset.

Figure 60. SIM_SOPT1 register—Rev. 1.x

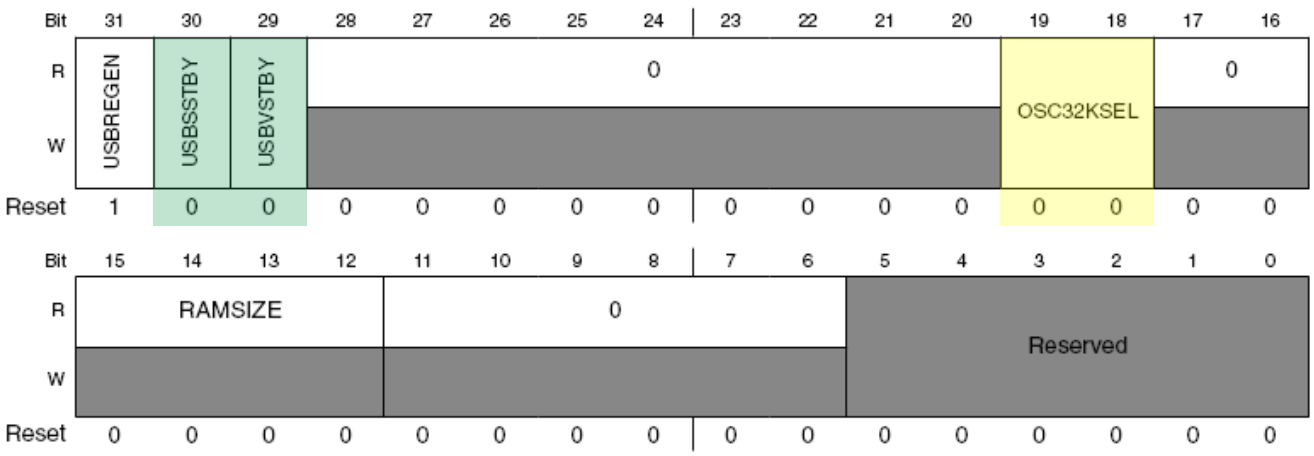


Figure 61. SIM_SOPT1 register—Rev. 2.x

New bits/fields added:

- USBSSTBY
- USBVSTBY

Remove bit/field names:

- USBSTBY

Changed bit/field names:

- MS → EZP_MS moved to the RCM_MR Register
- OSC32KSEL → Added bit to support option for LPO 1 kHz

Address: RCM_MR is 4007_F000h base + 7h offset = 4007_F007h

Bit	7	6	5	4	3	2	1	0
Read	0						EZP_MS	0
Write								
Reset	0	0	0	0	0	0	0	0

Figure 62. Mode register (RCM_MR)—Rev. 2.x

Address: SIM_SOPT1CFG is 4004_7000h base + 4h offset = 4004_7004h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0					USSWE	UVSWE	URWE	0								0						0		0								
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 63. SIM_SOPT1CFG—Rev. 2.x

Address: SIM_SOPT1CFG is 4004_7000h base + 4h offset = 4004_7004h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0					USSWE	UVSWE	URWE	0								0						0		0								
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 64. SIM_SOPT1CFG—Rev. 2.x

SIM_SOPT1CFG is a new register added.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0		SDHCSRC		0		I2SSRC		0		TIMESRC		0	USBSRC	0	PLLFLLSEL
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0			TRACECLKSEL	CMTUARTPAD	0	FBSL		0							MCGCLKSEL
W																
Reset	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0

Figure 65. SIM_SOPT2—Rev. 1.x

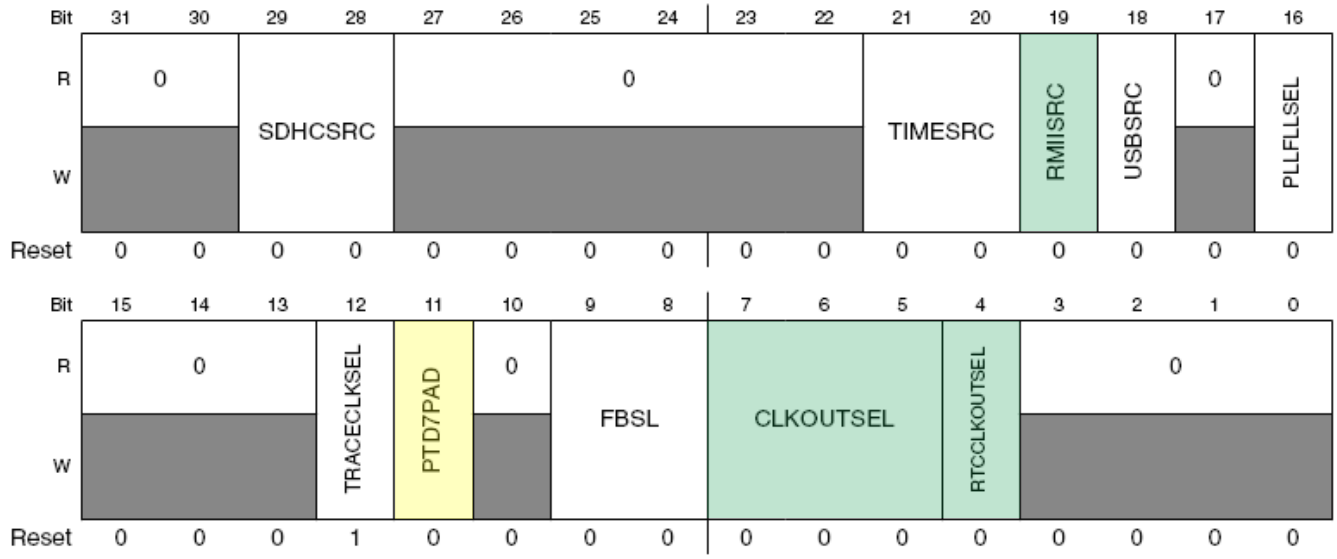


Figure 66. SIM_SOPT2—Rev. 2.x

New bits/fields added:

- RMIISRC
- CLKOUTSEL
- RTCCLKOUTTSEL

Remove bit/field names:

- I2SSRC

Changed bit/field names:

- CMTUARTPAD → PTD7PAD bit field renaming with no functional change

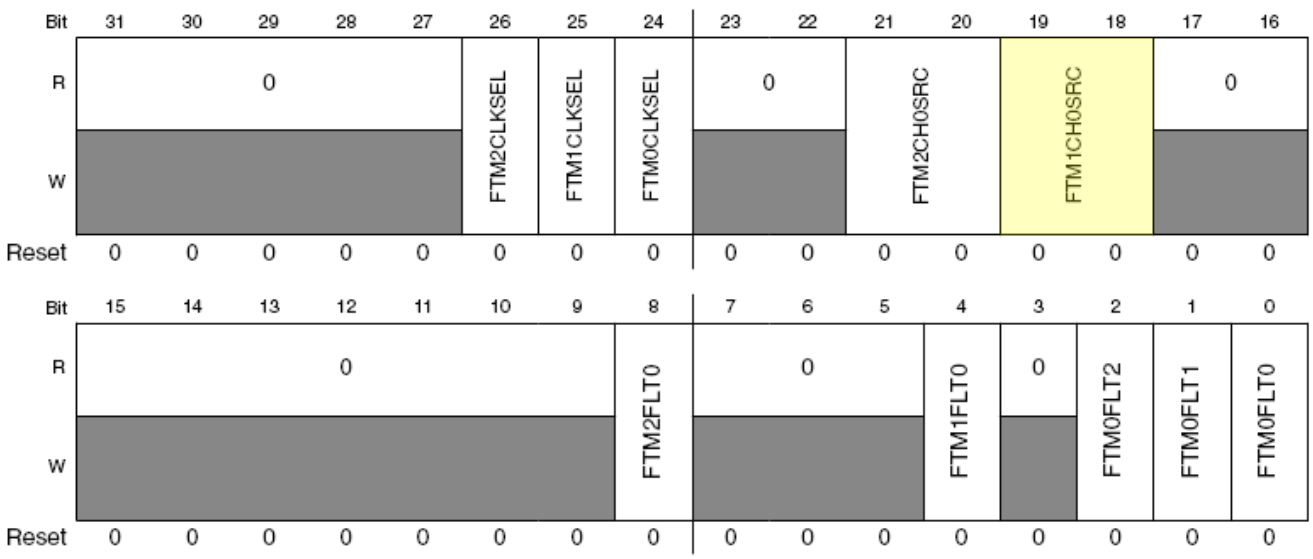


Figure 67. SIM_SOPT4—Rev. 1.x

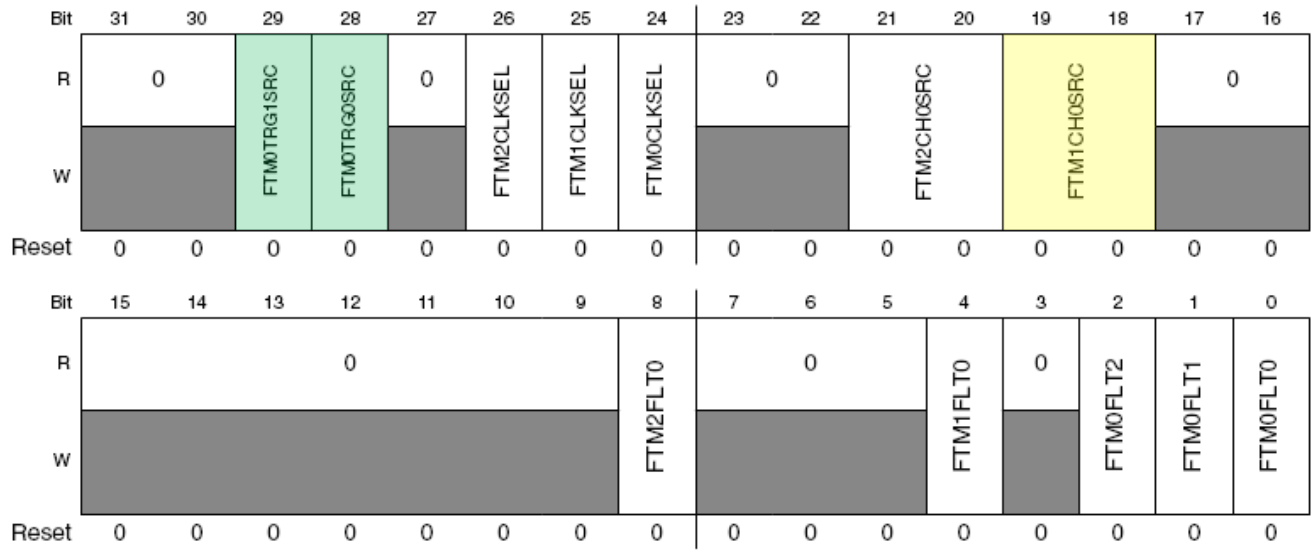


Figure 68. SIM_SOPT4—Rev. 2.x

New bits/fields added:

- FTM0TRG1SRC
- FTM0TRG0SRC

Changed bit/field names:

- FTM1CH0SRC → Added option for the USB Start-of-Frame Pulse

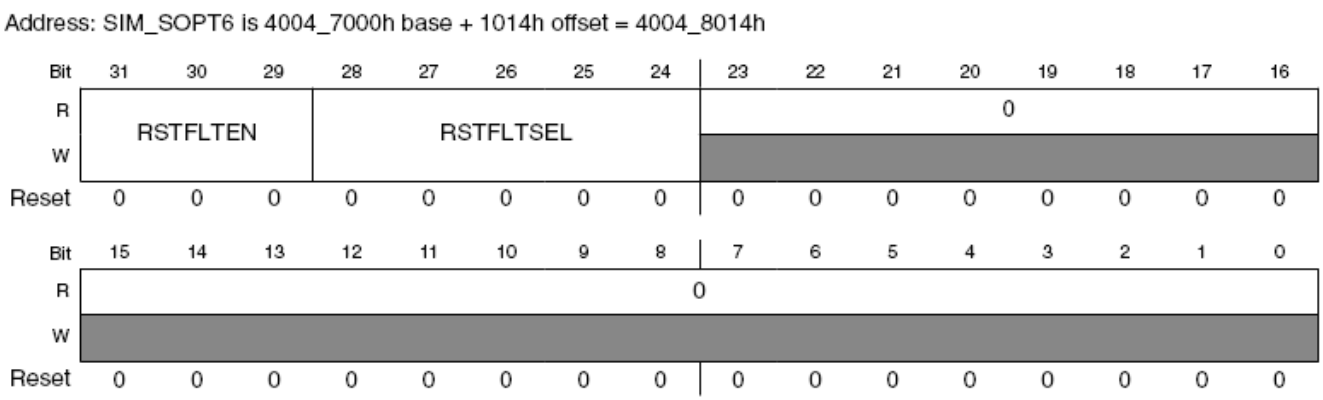


Figure 69. SIM_SOPT6—Rev. 1.x

Register removed and bit fields moved to RCM.

4.2 Multi-Clock Generator (MCG)

4.2.1 Impacted register

The MCG has two new registers, MCG_C7 and MCG_C8, and the MCG_ATC register has been renamed MCG_SC. In addition, many of the register field and bit names have been updated to match the naming convention used by the other modules.

The memory map changes are shown in the table below:

Memory map comparison				
	Kinetis Rev. 1.x		Kinetis Rev. 2.x	
	Location	Name	Location	Name
MCG Control 1 Register	4006_4000	MCG_C1	4006_4000	MCG_C1
MCG Control 2 Register	4006_4001	MCG_C2	4006_4001	MCG_C2
MCG Control 3 Register	4006_4002	MCG_C3	4006_4002	MCG_C3
MCG Control 4 Register	4006_4003	MCG_C4	4006_4003	MCG_C4
MCG Control 5 Register	4006_4004	MCG_C5	4006_4004	MCG_C5
MCG Control 6 Register	4006_4005	MCG_C6	4006_4005	MCG_C6
MCG Status Register	4006_4006	MCG_S	4006_4006	MCG_S
MCG Status and Control Register	4006_4008	MCG_ATC	4006_4008	MCG_SC
MCG Autorim Compare Value High Register	4006_400A	MCG_ATCVH	4006_400A	MCG_ATCVH
MCG Autorim Compare Value Low Register	4006_400B	MCG_ATCVL	4006_400B	MCG_ATCVL
MCG Control 7 Register	4006_400C	MCG_C7	4006_400C	MCG_C7
MCG Control 8 Register	4006_400D	MCG_C8	4006_400D	MCG_C8

The affected registers are summarized below.

4.2.1.1 MCG_C1 Register

Bit	7	6	5	4	3	2	1	0
Read / Write	CLKS		FRDIV			IREFS	IRCLKEN	IREFSTEN
Reset	0	0	0	0	0	1	0	0

Figure 70. Rev. 1.x

Bit	7	6	5	4	3	2	1	0
Read / Write	CLKS		FRDIV			IREFS	IRCLKEN	IREFSTEN
Reset	0	0	0	0	0	1	0	0

Figure 71. Rev. 2.x

There are no bit field changes in MCG_C1, but two new divide values have been added for FRDIV = 6 and FRDV = 7 in the case where MCG_C2[RANGE0] is greater than 0.

4.2.1.2 MCG_C2 Register

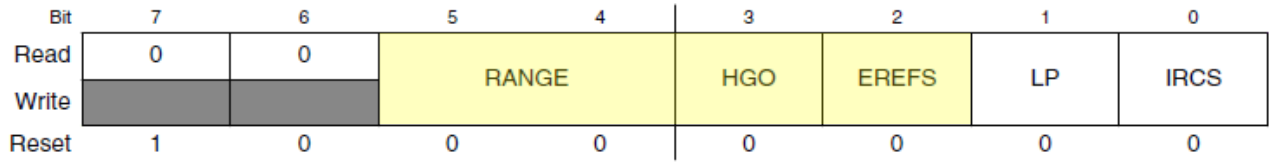


Figure 72. Rev. 1.x

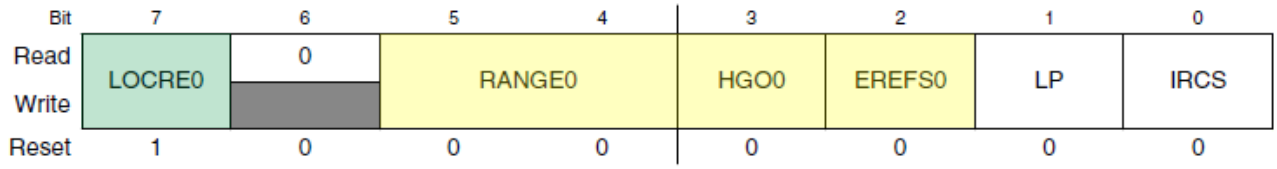


Figure 73. Rev. 2.x

New bits/fields added:

- LOCRE0

Changed bit/field names:

- RANGE → RANGE0
- HGO → HGO0
- EREFS → EREFS0

4.2.1.3 MCG_C5 Register

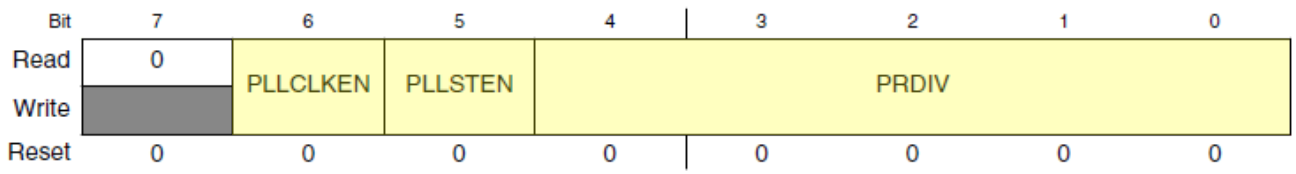


Figure 74. Rev. 1.x

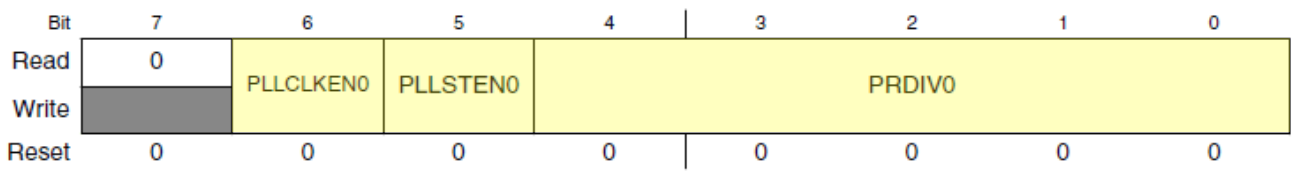


Figure 75. Rev. 2.x

Changed bit/field names:

- PLLCLKEN → PLLCLKEN0
- PLLSTEN → PLLSTEN 0
- PRDIV → PRDIV 0

4.2.1.4 MCG_C6

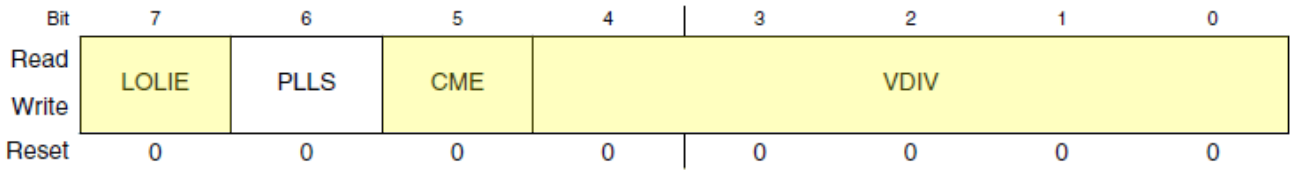


Figure 76. Rev. 1.x

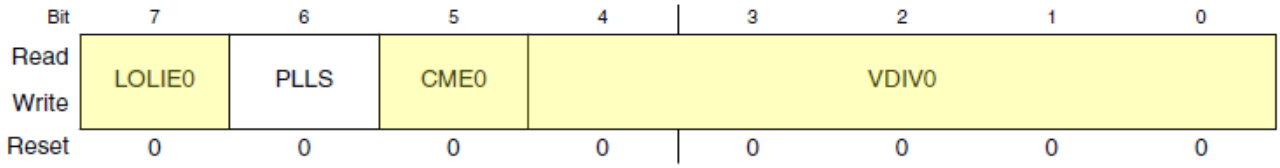


Figure 77. Rev. 2.x

Changed bit/field names:

- LOLIE → LOLIE 0
- CME → CME 0
- VDIV → VDIV 0

4.2.1.5 MCG_ATC Register

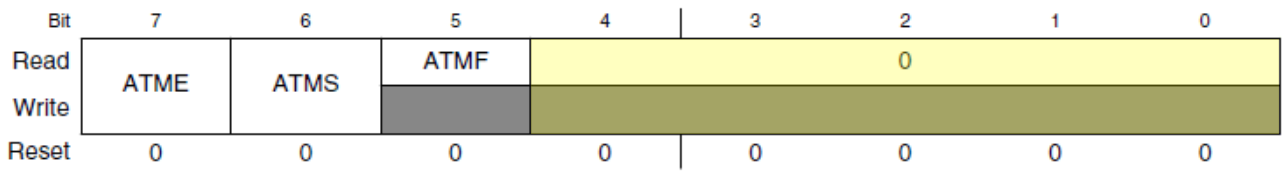


Figure 78. Rev. 1.x

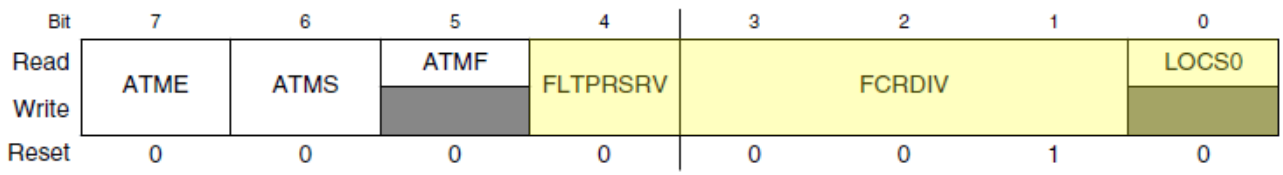


Figure 79. Rev. 2.x

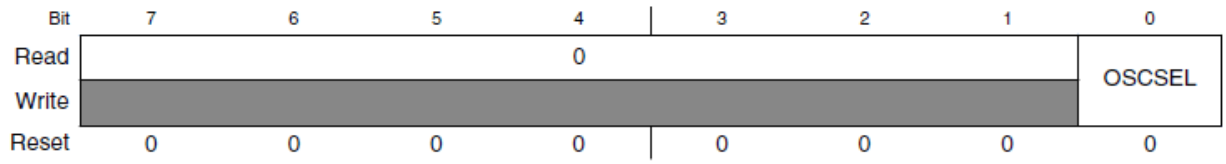
Register name is changed to MCG_SC

New bits/fields added:

- FLTPRSRV
- FCRDIV
- LOCS0

4.2.1.6 MCG_C7 Register

Address: MCG_C7 is 4006_4000h base + Ch offset = 4006_400Ch



New register

New bits/fields added:

- OSCSEL

4.2.1.7 MCG_C8 Register

Address: MCG_C8 is 4006_4000h base + Dh offset = 4006_400Dh

New register.

New bits/fields added:

- LOCRE1
- LOLRE
- CME1
- LOCS1

4.2.2 Features

The following new features have been included:

- Two additional FLL reference clock divider values of 1280 and 1536.

  Use case/Improvement: This allows the FLL reference clock frequency to be maintained within the specification range when using high speed an external high frequency clock.

- The addition of a clock monitor for the RTC, MCG_C8[CME1].

  Use case/Improvement: This provides a means of detecting whether the Real Time Clock (RTC) oscillator has stopped. It can also be used as a means of detecting whether the RTC oscillator has been initialized and is ready to use.

- The clock monitors have been improved to add the capability to select whether a reset or an interrupt is generated if a loss of clock occurs. A loss of clock flag has been added to indicate if a loss of clock event has occurred.

  Use case /Improvement: In cases where the external clock is not being used to provide the system clock, it may still be desirable to know if the clock has been lost but not reset the system. This addition allows an interrupt to be generated and flags to indicate which external clock has stopped.

- The fixed divide by 2 in the Fixed Frequency Clock (MCGFFCLK) has been removed.

Use case/Improvement: This allows the use of the FLL reference clock by the FTM Peripheral instead of half that frequency.

- The Fast Internal Reference Clock now has a programmable binary divider on the output.

  Use case/Improvement: This allows a wider range of clock frequencies to be used when the MCG is in FBI or BLPI modes, or when the fast IRC is selected as the MCGIRCLK for use by on-chip peripherals. This is particularly significant for VLPR where it is now possible to run the system at the maximum clock frequency using an internal clock.

- The mux selecting the system oscillator or the RTC oscillator as the FLL reference clock has been moved from the SIM to the MCG

  Use case/Improvement: When the RTC oscillator is being used as the FLL reference clock, it is now possible to also use the system oscillator to provide a clock to specific on-chip peripherals.

- The MCG registers can only be written to when the processor is in privileged mode.

  Use case/Improvement: This provides additional system-level security to prevent user mode applications from changing important system configuration settings.

- The DCO filter values can be preserved when the FLL is switched from an internally clocked mode (FBI or FEI) to an externally clocked mode (FBE or FEE) or vice versa.

  Use case/Improvement: In FLL modes, when switching between internal and external reference clocks, if the FLTPRSRV bit is set and the two reference clocks are at the same frequency, the FLL will remain locked and at the target frequency.

- The PLL loss of lock flag (LOLS) can now optionally generate an interrupt or a system reset if enabled. If a reset is generated, this source will be reflected in the RCM_SRS0 register.

  Use case/Improvement: This provides a means of resetting a system if the PLL falls out of lock to ensure that the system never executes software with an out-of-specification system clock frequency.

4.2.3 Software impact

The default configuration/operation of the MCG has not been changed. In terms of compatibility with MCG functionality, there are no software changes required. The one case where this is not true is if software is writing values to previously unused or reserved bit locations and these values are different from the new default values.

The main impact to software is the name changes associated with the registers and bit fields. The software will need to be changed to match the new names included in the new device header file.

If the RTC is being used as the reference clock for the FLL, or as the system clock, the selection of this clock is no longer made in the SIM_SOPT1; it is now in the MCG_C7 register.

Software will need to be added to make use of any new functionality. If the system makes use of the processor user mode, any MCG register configuration will need to be performed in privileged mode.

4.2.4 Hardware impact

There are no hardware changes required to maintain compatible with previous versions.

However, the change associated with the RTC oscillator selection has added the ability to also use the main system oscillator with a high frequency crystal to provide a separate clock to several on-chip peripherals.

4.3 Real Time Clock (RTC)

4.3.1 Impacted registers

The overall RTC memory map has no changes when comparing Rev. 1.x to Rev. 2.x; however, it does have a register bit field that has been implemented.

- RTC_IER[TSIE] Time Seconds Interrupt Enable bit and RTC_CR[WPE] Wakeup Pin Enable bit are now supported. Please refer to the Interrupt Vector Assignments section for the RTC Seconds Interrupt. Within the NVIC, the interrupt vector assignment is 83. Additionally, the RTC Seconds Interrupt can be used as a wakeup source to the LLWU module via the LLWU_M7IF pin.
- SIM_SOPT2[RTCCLKOUTSEL] RTC Clock Out Select bit selects between a 1 Hz clock or a 32.768 KHz clock on RTC_CLKOUT pin. Please refer to the SIM chapter for more details.

4.3.2 Features

The following new features have been included:

1. RTC_CLKOUT signal now has the option to be either 1 Hz or 32 kHz in all packages

   Use case/Improvement: The originally available option of 1 Hz allows the ability to accurately generate a 1 Hz clock even if the input crystal frequency deviates from the 32.768 kHz due to on-board variations. The 32.768 kHz option allows the input crystal frequency to be passed straight through and out to RTC_CLKOUT. This allows for other external devices, which require this clock, to use it as an input while leveraging the RTC oscillator and reducing the system BOM cost.

2. RTC_WAKEUP pin is now implemented to provide an external alarm event signal in some MAPBGA packages

   Use case/Improvement: This is an active low open drain pin that asserts when the MCU is powered down and the main RTC interrupt asserts. Note that this pin can be used to measure the RTC trigger output that monitors the MCU VDD level.

3. RTC seconds interrupt has been added

   Use case/Improvement: This is a dedicated interrupt that asserts every second. Note that it is an edge-triggered interrupt, so there is no flag to clear. This means the interrupt can be serviced without accessing the RTC registers, which decreases VBAT power consumption. The seconds interrupt can generate a wakeup from any low-power mode (Stop/VLPS/LLS/VLLS). For example, if the RTC Seconds signal is connected to the PDB, the PDB trigger can receive the RTC Seconds trigger input forcing ADC conversions in Run mode (where PDB is enabled).

4.3.3 Software impact

No software changes are required in order to execute existing code from Rev 1.x on Rev 2.x. To take advantage of additional features, some software changes would apply.

4.3.4 Hardware impact

No hardware changes are required in order to execute existing code from Rev 1.x on Rev 2.x. To take advantage of additional features, some hardware changes would apply.

4.4 Programmable Delay Block (PDB)

4.4.1 New Features

Change 1

Two additional pulse-out channels are added totaling three pulse-out channels.

Change 2

PDB Counter and DAC Interval Counter can now be paused when the processor is in Debug mode.

4.4.2 Impacted registers

Change 1

Two additional Pulse-Out n Delay registers (PDB0_PO1DLY and PDB0_PO2DLY) have been added. These two registers are used to set the sample window for CMP1 and CMP2 modules.

Change 2

The PDB Counter Register (PDBx_CNT) value should stop rolling when you halt the processor in Debug mode.

4.4.3 Memory map comparison

Change1

The table below shows the memory map differences between Rev. 1.x and Rev. 2.x devices.

Table 10. Memory map comparison

Location	Revision 1.x devices	Revision 2.x devices
0x40036198	N/A	Pulse-out delay register (PDB0_PO1DLY)
0x4003619C	N/A	Pulse-out delay register (PDB0_PO2DLY)

4.4.4 Software impact

Change 1

The additional PDB0_PO1DLY and PDB0_PO2DLY registers have been added so that you can have different window periods for CMP0, CMP1, and CMP2 outputs. If CMP0, CMP1, and CMP2 are configured as window mode in Rev. 1.x silicon, the registers' settings will be different if you want to make them work the same in Rev. 2.x silicon.

In Rev.1.x silicon, the Pulse-Out n Enable Register (PDBx_POEN) = 1 will enable the same value in PDB0_PO0DLY register for all the CMP0, CMP1, and CMP2.

In Rev. 2.x silicon, the Pulse-out n Enabled Register (PDBx_POEN) = 1 will only apply PDB0_PODDLY to CMP0. You will also need to set bit 1 of the PDBx_POEN register for CMP1 and set bit 2 of the PDBx_POEN register for CMP2. In Rev. 2.x silicon, the value assigned in PDB0_PO0DLY register only applies to CMP0. The PDB0_PO1DLY register value applies only to CMP1. PDB0_PO2DLY register value applies only to CMP2. To make your software work as it does with Rev. 1.x using the Rev. 2.x silicon, you will need to set PDB0_POEN to 0x07 and assign the same value to the PDB0_PO0DLY, PDB0_PO1DLY, and PDB0_PO2DLY registers.

4.4.5 Hardware impact

No hardware impact.

4.5 Flexible Timer Module (FTM)

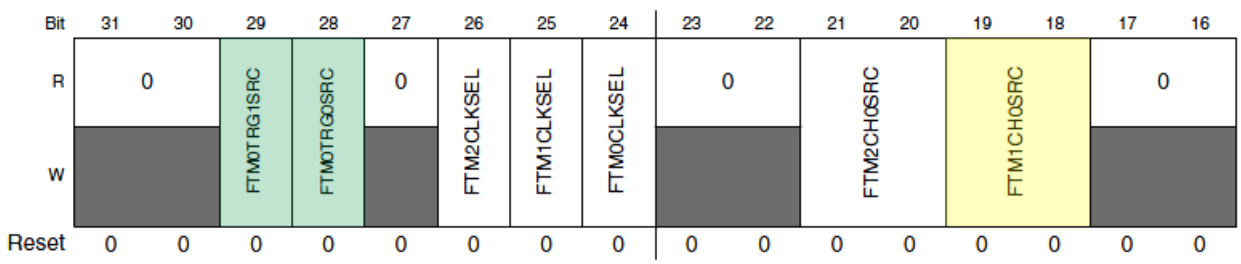
4.5.1 Impacted registers

Hardware trigger feature

- SIM_SOPT4 bit 28 for FTM1 match
- SIM_SOPT4 bit 29 for FTM2 match

Input capture feature

- SIM_SOPT4 bits [18-19] for USB start-of-frame



SIM_SOPT4 field descriptions

Field	Description
29 FTM0TRG1SRC	FlexTimer 0 Hardware Trigger 1 Source Select Selects the source of FTM0 hardware trigger 1. 0 PDB output trigger 1 drives FTM0 hardware trigger 1 1 FTM2 channel match drives FTM0 hardware trigger 1
28 FTM0TRG0SRC	FlexTimer 0 Hardware Trigger 0 Source Select Selects the source of FTM0 hardware trigger 0. 0 HSCMP0 output drives FTM0 hardware trigger 0 1 FTM1 channel match drives FTM0 hardware trigger 0
19–18 FTM1CH0SRC	FTM1 channel 0 input capture source select Selects the source for FTM1 channel 0 input capture. **NOTE:** When the FTM is not in input capture mode, clear this field. 00 FTM1_CH0 signal 01 CMP0 output 10 CMP1 output 11 USB start of frame pulse

Figure 80. SIM_SOPT4 register and field descriptions

4.5.2 Features

1. FTM peripheral adds FTM0 hardware synchronization trigger options from FTM1 and FTM2 match outputs.

   Use Case/Improvement: Makes it possible to connect two 16-bit FlexTimers, thus allowing you to increase PWM period with less dead time than using timer overflow flag (TOF) to connect both modules.

2. FTM peripheral adds FTM1 channel 0 input capture trigger option from USB start-of-frame pulse.

   Use Case/Improvement: Could be used to synchronize the MCU clock with the USB clock (needed for audio applications).

4.5.3 Software impact

No software changes are required to execute existing code from Rev 1.x on Rev 2.x. To take advantage of additional features, some software changes would apply. Only impacted registers need to be modified for specific use case.

4.5.4 Hardware impact

No hardware impact.

4.6 Low-power timer (LPTMR)

4.6.1 Impacted register

LPTMR_CNR

4.6.2 Features

The LPTMR has been updated to always ensure valid data when reading the LPTMR_CNR value.

4.6.3 Software impact

On each read of the LPTMR counter register, software must first write to the LPTMR counter register with any value. This will synchronize and register the current value of the LPTMR counter register into a temporary register. The contents of the temporary register are returned on each read of the LPTMR counter register.

4.6.4 Hardware impact

No hardware changes are required in order to execute existing code from Rev 1.x on Rev 2.x.

4.7 Universal asynchronous receiver/transmitters (UART)

4.7.1 Memory map comparison

There are a number of new registers added to the UART0 memory map to support new CEA709.1-B (LON) features. The new features and control of those features are all implemented as additions to the memory map, so previously existing registers and bits are unchanged.

The table below shows the memory map differences between revision 1.x devices and revision 2.x devices.

Table 11. Memory map comparison

Location	Rev. 1.x devices	Rev. 2.x devices
0x4006A021	N/A	UART CEA709.1-B Control Register 6 (UART0_C6)
0x4006A022	N/A	UART CEA709.1-B Packet Cycle Time Counter High (UART0_PCTH)
0x4006A023	N/A	UART CEA709.1-B Packet Cycle Time Counter Low (UART0_PCTL)
0x4006A024	N/A	UART CEA709.1-B Beta1 Timer (UART0_B1T)

Table continues on the next page...

Table 11. Memory map comparison (continued)

Location	Rev. 1.x devices	Rev. 2.x devices
0x4006A025	N/A	UART CEA709.1-B Secondary Delay Timer High (UART0_SDTH)
0x4006A026	N/A	UART CEA709.1-B Secondary Delay Timer Low (UART0_SDTL)
0x4006A027	N/A	UART CEA709.1-B Preamble (UART0_PRE)
0x4006A028	N/A	UART CEA709.1-B Transmit Packet Length (UART0_TPL)
0x4006A029	N/A	UART CEA709.1-B Interrupt Enable Register (UART0_IE)
0x4006A02A	N/A	UART CEA709.1-B WBASE (UART0_WB)
0x4006A02B	N/A	UART CEA709.1-B Status Register (UART0_S3)
0x4006A02C	N/A	UART CEA709.1-B Status Register (UART0_S4)
0x4006A02D	N/A	UART CEA709.1-B Received Packet Length (UART0_RPL)
0x4006A02E	N/A	UART CEA709.1-B Received Preamble Length (UART0_RPREL)
0x4006A02F	N/A	UART CEA709.1-B Collision Pulse Width (UART0_CPW)
0x4006A030	N/A	UART CEA709.1-B Receive Indeterminate Time (UART0_RIDT)
0x4006A031	N/A	UART CEA709.1-B Transmit Indeterminate Time (UART0_TIDT)

4.7.2 Features

Rev. 2.x adds support for the CEA709.1-B (LON) protocol on UART0. This protocol is commonly used in building automation and home networking applications. The features across the UART instantiations have not changed except for the addition of CEA709.1-B support.

4.7.3 Software impact

If your system does not need to take advantage of the new CEA709.1-B features, no software changes are required when migrating to Rev 2.x.

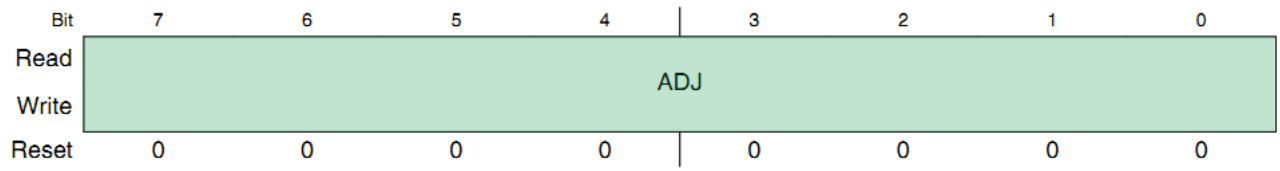
4.7.4 Hardware impact

If your system does not need to take advantage of the new CEA709.1-B features, no hardware changes are required when migrating to Rev 2.x.

4.8 Universal Serial Bus (USB)

4.8.1 Impacted register

Functionality of the USB voltage regulator can be controlled via the SIM_SOPT1CFG Register, a new register, and the SIM_SOPT1 Register, where bits and bit fields have been updated in order to support the new features. The USB has one new register: USBx_USBFRMADJUST.



USBx_USBFRMADJUST field descriptions

Figure 81. USBx USBFRMADJUST

4.8.2 Features

The following new features have been included:

- USB regulator Standby mode protection mechanism
- Start-of-frame (SOF) output on USB_SOF_OUT pin
- USB peripheral adds adjustable frame register in Host mode

4.8.2.1 USB regulator Standby mode

One of the features of the USB voltage regulator output Vout33 is to serve as the MCU's main power.

When Standby mode is enabled, the regulator limits maximum current load to 1 mA. Mask sets prior to 0N22D permit the configuration of the USB VREG into Standby mode while the MCU is in Run mode, which means that there won't be enough current for the MCU to execute instructions.

New masks implement a protection mechanism that controls when the USB voltage regulator is placed in Standby mode.

4.8.2.2 Start-of-frame output pin

Mask 0N22D includes a pin able to reflect the value of the SOF pin that can be used like a startup event for synchronization purposes (for audio, data loggers, and so on).

4.8.2.3 Adjustable frames in Host mode

SOF is normally generated every 12,000 12 MHz clock cycles. This new feature can adjust this by –128 to +127 to compensate for inaccuracies in the USB 48 MHz clock.

4.9 Software impact

4.9.1 USB regulator Standby mode

Two new bits have been added into the SIM_SOPT1 register to control the USB VREG Standby mode.

USBSSTBY—USB voltage regulator in Standby mode during Stop, VLPS, LLS, and VLLS modes.

USBVSTBY—USB voltage regulator in Standby mode during VLPR and VLPW modes.

The new SIM_SOPT1CFG register needs to allow writes prior to enabling the USB voltage regulator settings in SIM_SOPT1 (USBSSTB, USBVSTB, USBREGEN bits).

4.9.2 Start-of-frame output pin

The Pin Multiplexing section of this document provides more information about this pin.

4.9.3 Adjustable frames in Host mode

Write twos complement number to the new USBx_USBFRMADJUST register to adjust the period of USB frame.

4.10 Hardware impact

If your system does not need to take advantage of the new CEA709.1-B features, no hardware changes are required when migrating to Rev 2.x.

4.11 Comparator

4.11.1 Features added and changed

Change 1

A 12-bit DAC1 output has been added as part of the CMP0 input channel 4 (IN4) input.

The table below shows that in Rev. 2.x silicon, CMP0 input channel 4 is now connected to both the CMP0_IN4 external pin and the 12-bit DAC1 output. When CMP0 channel 4 is selected as an input, only one of the two signals should be enabled to avoid signal conflicts.

CMP0 inputs	Rev.1.x silicon	Rev. 2.x silicon
IN4	CMP0_IN4	12b DAC1 reference/CMP0_IN4

Change 2

The CMP0_IN4 input has been relocated to a different MCU pin.

The Pin Multiplexing section of this document shows that the CMP0_IN4 input has been moved to the new location on pin L4 for K40/K60 144 MAPBGA Kinetis devices or pin 39 for K40/K60 LQFP Kinetis devices. This new location may be different among different Kinetis family devices and package types. For the exact new pin location of the CMP0_IN4 input, please refer to the corresponding Kinetis Rev. 2.x reference manual and look for the Signal Multiplexing and Pin Assignments table.

Change 3

The Stop Mode Edge/Level Interrupt Control (SEMLB) bit in the CMP Status and Control Register (CMPx_SCR) is no longer available in Rev. 2.x silicon.

4.11.2 Software impacted

Change 1

The CMP0 IN4 input channel is now multiplexed with both the 12-bit DAC1 reference and CMP0_IN4 pin. Only one of the two signals should be enabled when CMP0 IN4 is selected. For example, if the CMP0 IN4 is intended for CMP0_IN4 pin, make sure you disable the 12-bit DAC1 output by DAC1_C0[DACEN] = 0. If CMP0 IN4 is intended for the 12-bit DAC1 output, the external source that drives the CMP0_IN4 pin should be disconnected.

Change 2

In Rev. 2.x silicon, PORTC_PCR_10[MUX] = 0 no longer selects CMP0_IN4 as an input. The new CMP0_IN4 input has been relocated to an analog-dedicated pin where all the multiplexed functions on this pin are analog functions and not digital GPIO. Such an analog-only pin does not require that you specify the pin function via a Pin Control Register. However, it is recommended that only one of the multiplexed analog functions should be enabled at the same pin.

4.11.3 Hardware impacted

Reroute the circuit to the new pin location for the CMP0_IN4 input. When the CMP0_IN4 pin is used, the corresponding multiplexed analog functions on the same pin should not be used and should be disabled.

4.12 Non-Maskable Interrupts (NMI)

4.12.1 Features

NMI adds two new features:

- A Flash Option Register (FOPT) bit that disables NMI pin function to avoid unwanted NMI interrupts and establish a specific function for this pin, since it is shared with $\overline{\text{EZP\_CS}}$.
- NMI pin assertion can generate an interrupt event from all power modes.

4.12.2 Impacted registers

The below figure shows the NMI_DIS bit added. This bit enables or disables the NMI function.

Bit Num	Field	Value	Definition
7-3	Reserved		Reserved for future expansion.
2	NMI_DIS	0	NMI interrupts are always blocked. The associated pin continues to default to NMI pin controls with internal pullup enabled.
		1	NMI pin/interrupts reset default to enabled.
1	EZPORT_DIS	0	EzPort operation is disabled. The device always boots to normal CPU execution and the state of EZP_CS signal during reset is ignored. This option avoids inadvertent resets into EzPort mode if the EZP_CS/NMI pin is used for its NMI function.
		1	EzPort operation is enabled. The state of EZP_CS pin during reset determines if device enters EzPort mode.
0	LPBOOT	0	Low-power boot: OUTDIVx values in SIM_CLKDIV1 register are auto-configured at reset exit for higher divide values that produce lower power consumption at reset exit. • Core and system clock divider (OUTDIV1) and bus clock divider (OUTDIV2) are 0x7 (divide by 8) • Flash clock divider (OUTDIV4) and FlexBus clock divider (OUTDIV3) are 0xF (divide by 16)
		1	Normal boot: OUTDIVx values in SIM_CLKDIV1 register are auto-configured at reset exit for higher frequency values that produce faster operating frequencies at reset exit. • Core and system clock divider (OUTDIV1) and bus clock divider (OUTDIV2) are 0x0 (divide by 1) • Flash clock divider (OUTDIV4) and FlexBus clock divider (OUTDIV3) are 0x1 (divide by 2)

4.12.3 Software impact

No required software changes. To take advantage of additional features, some software changes would apply.

4.12.4 Hardware impact

No hardware impact with these new features.

4.13 Periodic interrupt timers (PIT)

4.13.1 Impacted registers

Although the overall PIT memory map has not changed between Rev. 1.x and Rev. 2.x, it does have a new register bit field added to the Timer Control Register.

The affected registers are summarized below.

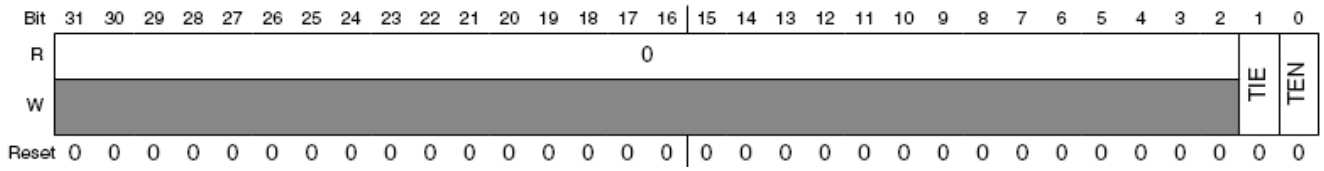


Figure 82. PIT_TCTRLn Register Rev. 1.x

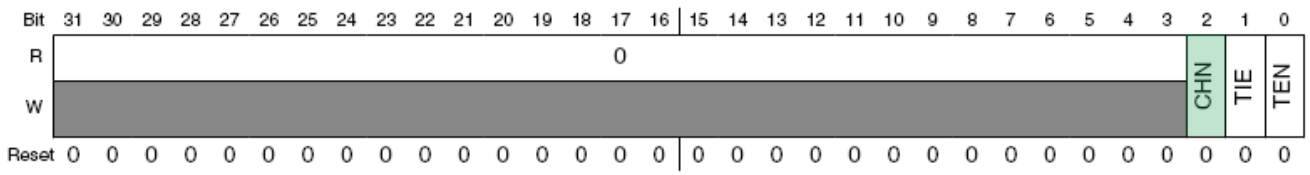


Figure 83. PIT_TCTRLn Register Rev. 2.x

New bits/fields added: CHN

4.13.2 Features

The following new feature has been included:

- The ability for the peripheral to concatenate timer channels via the Chain Mode bit, PIT_TCTRLn[CHN].

  Use case/Improvement: This provides a means of chaining a timer to a previous timer. For example, if this bit is set for channel 2, then timer 2 is chained to timer 1, thus creating a 64-bit counter.

4.13.3 Software impact

No software changes are required to execute existing code from Rev. 1.x on Rev. 2.x. To take advantage of additional features, some software changes would apply.

4.13.4 Impacted registers

Although the overall PIT memory map has not changed between Rev. 1.x and Rev. 2.x, it does have a new register bit field added to the Timer Control Register.

The affected registers are summarized below.

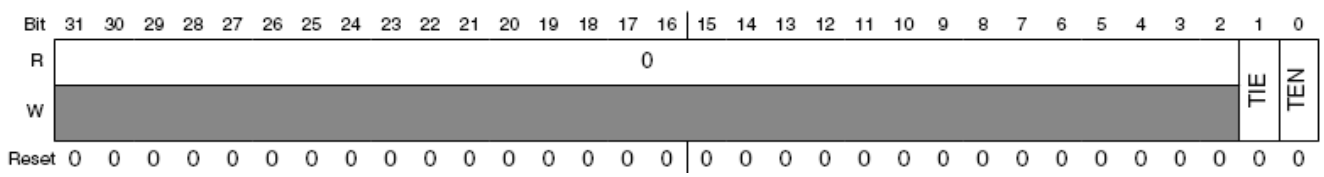


Figure 84. PIT_TCTRLn Register Rev. 1.x

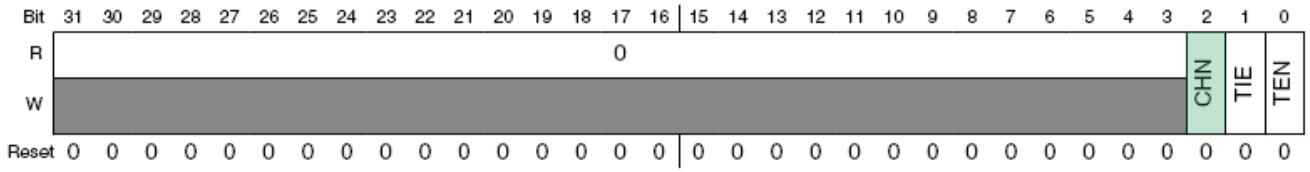


Figure 85. PIT_TCTRLn Register Rev. 2.x

New bits/fields added: CHN

4.14 Ethernet MAC (ENET)

4.14.1 Impacted registers

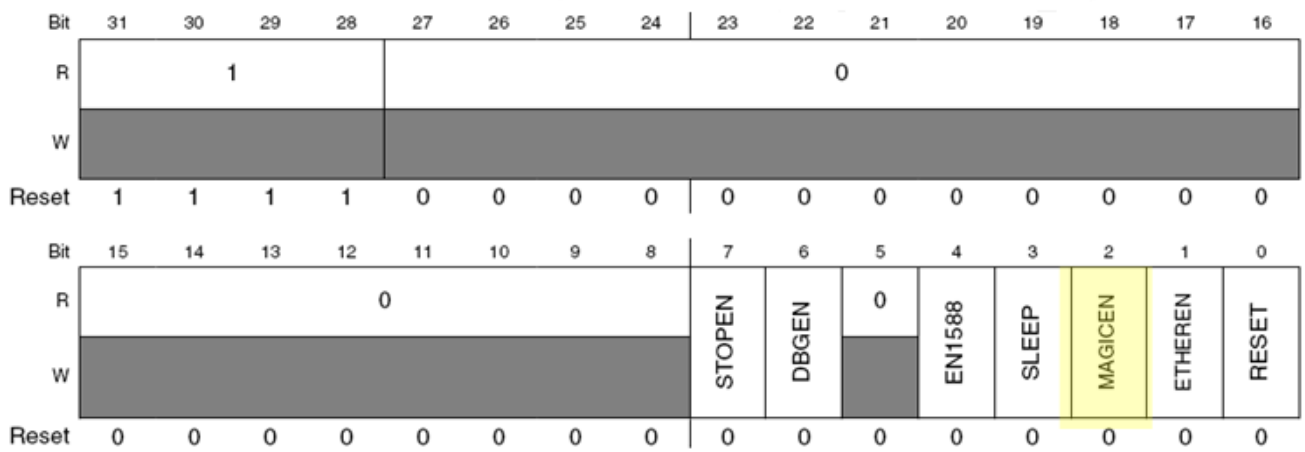
The ENET adds two new features, ENET_ECR[DBSWP] and enhanced RxBD[VPCP], and changed functionality of ENET_ECR[MAGICEN].

There are no memory map changes but mentioned registers are included for reference as shown in the table below.

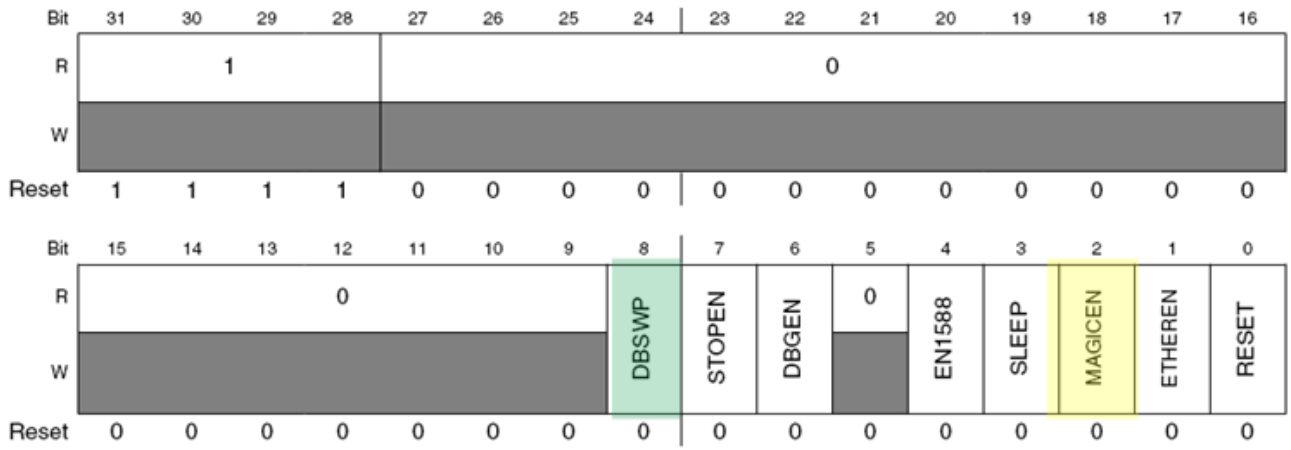
Memory map comparison				
	Kinetis Rev. 1.x		Kinetis Rev. 2.x	
	Location	Name	Location	Name
Ethernet Control Register	400C_0024	ENET_ECR	400C_0024	ENET_ECR
Enhanced Rx Buffer Descriptor	N/A	Enhanced RxBD	N/A	Enhanced RxBD

4.14.1.1 ENET_ECR Register

Revision 1.x



Revision 2.0:



New bits/fields added: DBSWP

New bits/fields changed: MAGICEN

4.14.1.2 Enhanced Rx Buffer Descriptor (RxBD)

Revision 1.x:

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Offset + 0	E	RO1	W	RO2	L	—	—	M	BC	MC	LG	NO	—	CR	OV	TR
Offset + 2	Data length															
Offset + 4	Rx data buffer pointer - A[31:16]															
Offset + 6	Rx data buffer pointer - A[15:0]															
Offset + 8	ME	—	—	—	—	PE	CE	UC	INT	—	—	—	—	—	—	—
Offset + A	—	—	—	—	—	—	—	—	—	—	ICE	PCR	—	VLAN	IPV6	FRAG
Offset + C	Header length					—	—	—	Protocol type							
Offset + E	Payload checksum															
Offset + 10	BDU	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Offset + 12	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Offset + 14	1588 timestamp [31:16]															
Offset + 16	1588 timestamp [15:0]															
Offset + 18	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Offset + 1A	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Offset + 1C	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Offset + 1E	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Revision 2.0:

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Offset + 0	E	RO1	W	RO2	L	—	—	M	BC	MC	LG	NO	—	CR	OV	TR
Offset + 2	Data length															
Offset + 4	Rx data buffer pointer - A[31:16]															
Offset + 6	Rx data buffer pointer - A[15:0]															
Offset + 8	ME	—	—	—	—	PE	CE	UC	INT	—	—	—	—	—	—	—
Offset + A	VPCP			—	—	—	—	—	—	—	ICE	PCR	—	VLAN	IPV6	FRAG
Offset + C	Header length					—	—	—	Protocol type							
Offset + E	Payload checksum															
Offset + 10	BDU	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Offset + 12	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Offset + 14	1588 timestamp [31:16]															
Offset + 16	1588 timestamp [15:0]															
Offset + 18	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Offset + 1A	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Offset + 1C	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Offset + 1E	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

New bits/fields added: VPCP

4.14.1.3 Features

The following new features have been included:

1. Compliant with AMD magic packet detection

   Use case/Improvement: The previous ENET version decodes a magic packet if the frame is formed with a synchronization stream (six consecutive 0xFF bytes) followed by sequence of six consecutive unicast MAC addresses of the node to be awakened. These 42 bytes must not have anything between them and might be placed anywhere in the Ethernet frame. Then it can be mounted over Ethernet, UDP, IP, TCP, or any other protocol.

   The new version is compliant with AMD magic packet detection, which requires a frame formed with a synchronization stream (six consecutive 0xFF bytes), followed by sequence of sixteen consecutive unicast MAC addresses of the node to be awakened. These 102 bytes might be placed anywhere in the Ethernet frame.

2. Addition of VLAN Priority bits to the enhanced receive buffer descriptor

   Use case/Improvement: When using enhanced buffer descriptors, the RxBD[VPCP] (VLAN priority code point) detects the VLAN frame priority level in three bits (only valid when the RxBD[L] bit is set). This value can be used to prioritize different classes of traffic (for example, voice, video, or data) where a value of 7 is the highest priority VLAN frame.

3. Addition of buffer descriptor endianess option

   Use case/Improvement: When writing the buffer descriptors fields, regardless of the enhanced or legacy mode, an endianess software conversion from big endian (register endianess) to little endian is required in previous and new MAC-NET versions. When reading the buffer descriptor fields, a little endian to big endian conversion is also required.

However, ENET_ECR[DBSWP] allows managing buffer descriptors in ARM Cortex-M4 native little endian, performing the endianess conversion in hardware.

4.14.1.4 Software impact

The default configuration/operation of the ENET has not been changed. All the new features make the ENET compliant with a standard or offload work to the application, giving it to the hardware. Two of the three features are back-compatible with previous implementations and don't require any upgrade in software. The addition of buffer descriptor (BD) endianess is turned off by default and the corresponding bit in Kinetis Rev. 2.x is reserved in Kinetis Rev. 1.x. A change is only required if it is explicitly enabled.

- Compliant with AMD magic packet detection

  The new version is compliant with AMD magic packet detection. It requires a frame formed with a synchronization stream (six consecutive 0xFF bytes) followed by sequence of sixteen consecutive unicast MAC addresses of the node to be awakened. These 102 bytes might be placed anywhere in the Ethernet frame. Six consecutive unicast MAC addresses no longer wake up the MCU.

- Addition of VLAN Priority bits to the enhanced receive buffer descriptor

  The application no longer needs to parse the VLAN packet to get the frame priority. It can be taken from the enhanced RxBD[VPCP] field. Note that enhanced buffer descriptor mode must be enabled.

- Addition of buffer descriptor endianess option
  Before implementing the buffer descriptor endianess feature, a legacy buffer descriptor is exposited in the following C-language structure.

```
typedef struct
{
   uint16_t status;       /* control and status */
   uint16_t length;       /* transfer length */
   uint8_t *data;         /* buffer address */
} NBUF;
```

  All the elements of the structure must be written in big endian; a 16-bit (status and length) and a 32-bit (pointer to data) endianess conversion is required in software when accessing elements of the structure that describe a buffer descriptor. The same applies for read accesses.

  For Kinetis Rev. 2.x, if buffer the descriptor endianess feature is enabled, then the following C-language structure describes a buffer descriptor.

```
typedef struct
{
   uint16_t length;       /* transfer length */
   uint16_t status;       /* control and status */
   uint8_t *data;         /* buffer address */
} NBUF;
```

  Then, all the elements of the structure can be accessed in native little endian.

  Note the change between length and status elements because the hardware endianess is applied in 32-bit accesses and not in 16-bit accesses.

  Legacy buffer descriptors are used in this example instead of enhanced BD for simplicity, but the same apply to both modes.

4.14.1.5 Hardware impact

There are no hardware changes required to be compatible with previous versions.

4.15 Voltage reference (VREF)

4.15.1 Impacted registers

No registers have been added to or removed from the VREF module memory map. There have been changes to the bit fields within the registers; these are summarized below.

4.15.1.1 VREF_TRM register

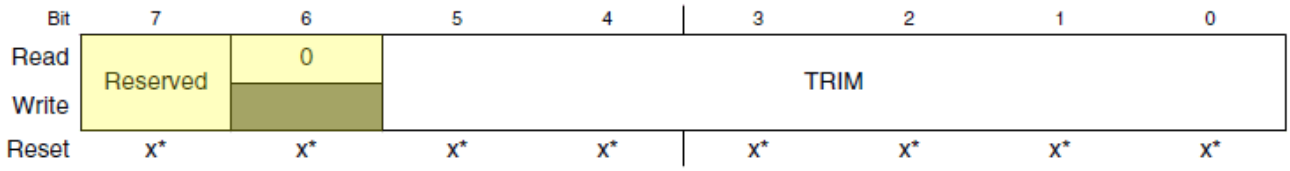


Figure 86. Rev. 1.x

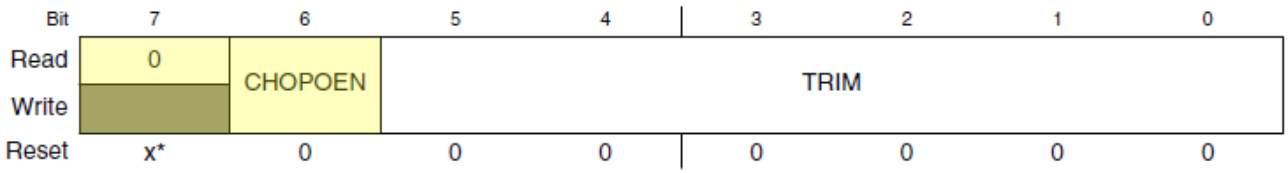


Figure 87. Rev. 2.x

New fields/bits added: CHOPOEN

Changed field/bit names: Bit 7 is now a read-only bit

4.15.1.2 VREF_SC register

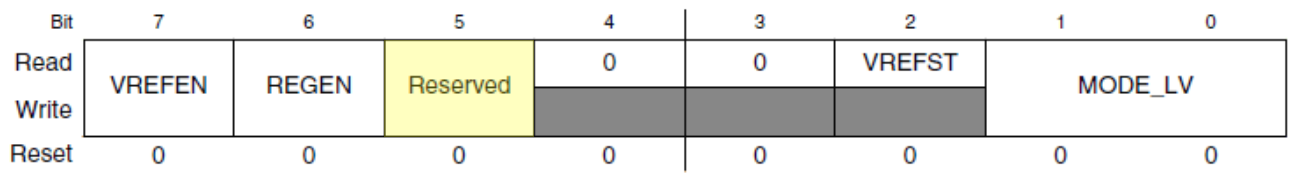


Figure 88. Rev. 1.x

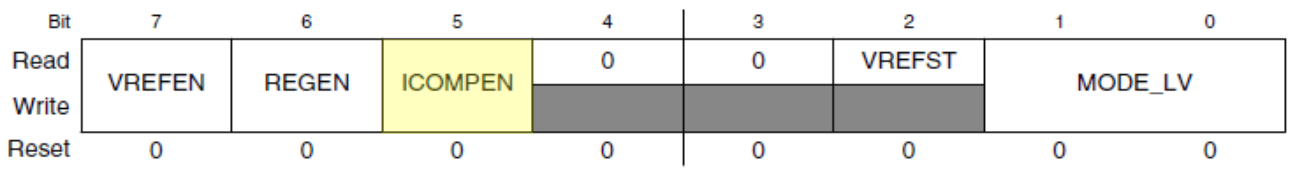


Figure 89. Rev. 2.x

New fields/bits added: ICOMPEN

4.15.2 Features

The following new features have been included:

- A chopping circuit has been added.

  Use case/Improvement: Enabling this option will improve the temperature coefficient and voltage drift of the VREF output voltage.

- Two output buffer modes are now available, including a new low-power buffer.

  Use case/Improvement: In applications where the VREF is being used as an external reference and the loading of the reference is very small (< 0.5 mA), this provides a lower power option when using the VREF voltage.

- Additional temperature compensation has been provided.

  Use case/Improvement: Provides optimum temperature coefficient of the VREF voltage.

- It is possible to enable the internal VREF voltage regulator when the device is in VLPR. This option is actually enabled by setting the BGEN bit in the PMC module.

  Use case/Improvement: Using the internal VREF voltage regulator provides additional supply noise rejection and reduces the drift of the VREF voltage due to VDD variation. This provides a more accurate and consistent reference voltage when in VLPR.

4.15.3 Software impact

In order to optimize VREF performance, the VREF_TRM[CHOPOEN] and VREF_SC[ICOMPEN] bits will need to be programmed appropriately. The exact configuration will not be determined until Rev. 2.x has been evaluated. This configuration will be required for the VREF module to meet data sheet values.

Previously, a read-modify-write operation was required to update the VREF_TRM register. This is no longer required as bit 7 of this register is now a read-only bit.

In order to make use of the new functionality, software would need to be added.

4.15.4 Hardware impact

There are no hardware changes associated with the VREF module.

5 Appendices

5.1 Pin Multiplexing

The table below uses the following conventions:

- **(+)**—added from Rev. 1.x to Rev. 2.x

- **(-)**—removed from Rev. 1.x to Rev. 2.x
- **Shadedregion**—changed from Rev. 1.x to Rev. 2.x

K60/K40 144 MAPBGA	K40 144 LQFP	K60 144 LQFP	ALT0	ALT1	ALT2	ALT3	ALT4	ALT5	ALT6	ALT7
L5	–	–	(+) RTC_WAKEUP_b							
D3	1	1								(+) RTC_CLKOUT
D2	2	2								(+) SPI1_SIN
E4	4	4								(+) SPI1_SOUT
E1	9	9							(-) I2S0_CLKIN	(+) USB_SOF_OUT
F4	10	10					I2S0_RXD →I2S0_RXD0			
F3	11	11			(+) I2S0_RXD1					
F2	12	12			(+) I2S0_TXD1					
F1	13	13					I2S0_TXD →I2S0_TXD0			
L4	39	39	(+) CMP0_IN4							
K4	47	47			(+) ENET_1588_CLKIN		(-) ENET_1588_CLKIN			
J5	50	50			UART0_CTS_b → UART0_CTS_b/ UART0_COL_b					
M8	55	55			(+) USB_CLKIN				I2S0_RX_BCLK → I2S0_TX_BCLK	

Table continues on the next page...

K9	64	64							I2S0_TXD → I2S0_TXD0	
L10	66	66							I2S0_TX_BCLK → I2S0_RX_BCLK	(+) I2S0_TXD1
L11	67	67							I2S0_RXD → I2S0_RXD0	
K10	68	68				UART0_CTS_b →UART0_CTS_b/ UART0_COL_b				(+) I2S0_RXD1
K11	69	69								(-) I2S0_CLKIN
M11	73	73							LPT0ALT1 → LPTMR0_ALT1	
G11	84	84				UART0_CTS_b → UART0_CTS_b/ UART0_COL_b				
B12	103	103					(-) I2S0_TXD		(+) I2S0_TXD1	
B11	104	104							(+) I2S0_TXD0	
A12	105	105							(+) I2S0_TX_FS	
A11	106	106						FB_CLKOUT → CLKOUT	(+) I2S0_TX_BCLK	
D8	114	110				(+) LPTMR0_ALT2	LPTMR0_ALT2 → I2S0_RXD0			
C8	115	111					(+) I2S0_RX_BCLK		(+) I2S0_MCLK	
B8	116	112				(+) USB_SOF_OUT	(+) I2S0_RX_FS			

Table continues on the next page...

A8	117	113				(-) I2S0_MCLK	I2S0_CLKIN → I2S0_MCLK			
C7	119	115	(-) CMP0_IN4							
B7	120	116					I2S0_RXD → I2S0_RXD1			
A3	134	134				UART0_CTS_b → UART0_CTS_b/ UART0_COL_b				

5.2 Memory Map

5.2.1 Kinetis Rev. 1.x memory map

System 32-bit Address Range	Destination Slave	Access
0x0000_0000–0x0FFF_FFFF	Program flash and read-only data (Includes exception vectors in first 1024 bytes)	All masters
0x1000_0000–0x13FF_FFFF	• For MK60DN256ZVLQ10: Reserved • For MK60DX256ZVLQ10: FlexNVM • For MK60DN512ZVLQ10: Reserved • For MK60DN256ZVMD10: Reserved • For MK60DX256ZVMD10: FlexNVM • For MK60DN512ZVMD10: Reserved	All masters
0x1400_0000–0x17FF_FFFF	For devices with FlexNVM: FlexRAM For devices with program flash only: Programming acceleration RAM	All masters
0x1800_0000–0x1FFF_FFFF	SRAM_L: Lower SRAM (ICODE/DCODE)	All masters
0x2000_0000–0x200F_FFFF	SRAM_U: Upper SRAM bitband region	All masters
0x2010_0000–0x21FF_FFFF	Reserved	–
0x2200_0000–0x23FF_FFFF	Aliased to SRAM_U bitband	Cortex-M4 core only
0x2400_0000–0x3FFF_FFFF	Reserved	–

5.2.2 Kinetis Rev. 2.x memory map

System 32-bit Address Range	Destination Slave	Access
0x0000_0000–0x07FF_FFFF	Program flash and read-only data (Includes exception vectors in first 1024 bytes)	All masters
0x0800_0000–0x0FFF_FFFF	FlexBus (Aliased area)	Cortex-M4 core (M0) only
0x1000_0000–0x13FF_FFFF	• For MK60DN256VLQ10: Reserved • For MK60DX256VLQ10: FlexNVM • For MK60DN512VLQ10: Reserved • For MK60DN256VMD10: Reserved • For MK60DX256VMD10: FlexNVM • For MK60DN512VMD10: Reserved	All masters
0x1400_0000–0x17FF_FFFF	For devices with FlexNVM: FlexRAM For devices with program flash only: Programming acceleration RAM	All masters
0x1800_0000–0x1BFF_FFFF	FlexBus (Aliased area)	Cortex-M4 core (M0) only
0x1C00_0000–0x1FFF_FFFF	SRAM_L: Lower SRAM (ICODE/DCODE)	All masters
0x2000_0000–0x200F_FFFF	SRAM_U: Upper SRAM bitband region	All masters
0x2010_0000–0x21FF_FFFF	Reserved	–

Added FlexBus aliased memory region with 0 wait state access from Cortex-M4 core

5.3 Interrupt channel assignments

The interrupt source assignments are defined in the following table.

For Kinetis Light - K20-512, it would be nice to have the vector numbers match the P2 implementation as shown in the below table. If there is a significant Gate Count cost associated with this configuration, then the vector numbers should be made sequential.

- Vector number — the value stored on the stack when an interrupt is serviced.
- IRQ number — non-core interrupt source count, which is the vector number minus 16.

The IRQ number is used within ARM's NVIC documentation.

Table 15. Interrupt vector assignments

Address	Vector	IRQ[1]	NVIC non-IPR register number [2]	NVIC IPR register number [3]	Source module	Source description
ARM Core System Handler Vectors						
0x0000_0000	0	–	–	–	ARM core	Initial Stack Pointer
0x0000_0004	1	–	–	–	ARM core	Initial Program Counter
0x0000_0008	2	–	–	–	ARM core	Non-maskable Interrupt (NMI)

Table continues on the next page...

Table 15. Interrupt vector assignments (continued)

Address	Vector	IRQ[1]	NVIC non-IPR register number [2]	NVIC IPR register number [3]	Source module	Source description
0x0000_000C	3	–	–	–	ARM core	Hard Fault
0x0000_0010	4	–	–	–	ARM core	MemManage Fault
0x0000_0014	5	–	–	–	ARM core	Bus Fault
0x0000_0018	6	–	–	–	ARM core	Usage Fault
0x0000_001C	7	–	–	–	—	—
0x0000_0020	8	–	–	–	—	—
0x0000_0024	9	–	–	–	—	—
0x0000_0028	10	–	–	–	—	—
0x0000_002C	11	–	–	–	ARM core	Supervisor call (SVCall)
0x0000_0030	12	–	–	–	ARM core	Debug Monitor
0x0000_0034	13	–	–	–	—	—
0x0000_0038	14	–	–	–	ARM core	Pendable request for system service (PendableSrvReq)
0x0000_003C	15	–	–	–	ARM core	System tick timer (SysTick)
Non-Core Vectors						
On-platform Vectors						
0x0000_0040	16	0	0	0	DMA	DMA channel 0, 16 transfer complete
0x0000_0044	17	1	0	0	DMA	DMA channel 1, 17 transfer complete
0x0000_0048	18	2	0	0	DMA	DMA channel 2, 18 transfer complete
0x0000_004C	19	3	0	0	DMA	DMA channel 3, 19 transfer complete
0x0000_0050	20	4	0	1	DMA	DMA channel 4, 20 transfer complete
0x0000_0054	21	5	0	1	DMA	DMA channel 5, 21 transfer complete
0x0000_0058	22	6	0	1	DMA	DMA channel 6, 22 transfer complete
0x0000_005C	23	7	0	1	DMA	DMA channel 7, 23 transfer complete
0x0000_0060	24	8	0	2	DMA	DMA channel 8, 24 transfer complete
0x0000_0064	25	9	0	2	DMA	DMA channel 9, 25 transfer complete
0x0000_0068	26	10	0	2	DMA	DMA channel 10, 26 transfer complete
0x0000_006C	27	11	0	2	DMA	DMA channel 11, 27 transfer complete
0x0000_0070	28	12	0	3	DMA	DMA channel 12, 28 transfer complete
0x0000_0074	29	13	0	3	DMA	DMA channel 13, 29 transfer complete
0x0000_0078	30	14	0	3	DMA	DMA channel 14, 30 transfer complete
0x0000_007C	31	15	0	3	DMA	DMA channel 15, 31 transfer complete
0x0000_0080	32	16	0	4	DMA	DMA error interrupt channels 0-1531
0x0000_0084	33	17	0	4	MCM	Normal interrupt
Off-platform Vectors						
0x0000_0088	34	18	0	4	Flash memory	Command complete
0x0000_008C	35	19	0	4	Flash memory	Read collision

Table continues on the next page...

Table 15. Interrupt vector assignments (continued)

Address	Vector	IRQ[1]	NVIC non-IPR register number [2]	NVIC IPR register number [3]	Source module	Source description
0x0000_0090	36	20	0	5	Mode Controller	Low-voltage detect, low-voltage warning
0x0000_0094	37	21	0	5	LLWU	Low Leakage Wakeup **NOTE:** The LLWU interrupt must not be masked by the interrupt controller to avoid a scenario where the system does not fully exit stop mode on an LLS recovery.
0x0000_0098	38	22	0	5	WDOG	WDOG | EWM
0x0000_009C	39	23	0	5	—RNG	—Randon Number Generator
0x0000_00A0	40	24	0	6	I^2C0	—
0x0000_00A4	41	25	0	6	—I^2C1	—
0x0000_00A8	42	26	0	6	SPI0	Single interrupt vector for all sources
0x0000_00AC	43	27	0	6	—SPI1	—Single interrupt vector for all sources
0x0000_00B0	44	28	0	7	—I^2S0	——Transmit
0x0000_00B4	45	29	0	7	—I^2S0	——Receive
0x0000_00B8	46	30	0	7	—UART0	—Single interrupt vector for UART LON sources
0x0000_00BC	47	31	0	7	UART0	Single interrupt vector for UART status sources
0x0000_00D0	48	32	1	8	UART0	Single interrupt vector for UART error sources
0x0000_00D4	49	33	1	8	UART1	Single interrupt vector for UART status sources
0x0000_00D8	50	34	1	8	UART1	Single interrupt vector for UART error sources
0x0000_00DC	51	35	1	8	UART2	Single interrupt vector for UART status sources
0x0000_00E0	52	36	1	9	UART2	Single interrupt vector for UART error sources
0x0000_00E4	53	37	1	9	UART3	Single interrupt vector for UART status sources
0x0000_00E8	54	38	1	9	UART3	Single interrupt vector for UART error sources
0x0000_00EC	55	39	1	9	ADC0	—
0x0000_00F0	56	40	1	10	CMP0	—
0x0000_00F4	57	41	1	10	CMP1	—
0x0000_00F8	58	42	1	10	FTM0	Single interrupt vector for all sources
0x0000_00FC	59	43	1	10	—FTM1	—Single interrupt vector for all sources
0x0000_0100	60	44	1	11	—FTM2	—Single interrupt vector for all sources
0x0000_0104	61	45	1	11	CMT	—

Table continues on the next page...

Table 15. Interrupt vector assignments (continued)

Address	Vector	IRQ[1]	NVIC non-IPR register number [2]	NVIC IPR register number [3]	Source module	Source description
0x0000_0108	62	46	1	11	RTC	Alarm interrupt
0x0000_010C	63	47	1	11	—RTC	—Seconds interrupt
0x0000_0110	64	48	1	12	PIT	Channel 0
0x0000_0114	65	49	1	12	PIT	Channel 1
0x0000_0118	66	50	1	12	PIT	Channel 2
0x0000_011C	67	51	1	12	PIT	Channel 3
0x0000_0120	68	52	1	13	PDB	—
0x0000_0124	69	53	1	13	—USB OTG	—
0x0000_0128	70	54	1	13	—USB Charger Detect	—
0x0000_012C	71	55	1	13	—DryIce	—Tamper
0x0000_012C	71	55	1	13	—	—
0x0000_0130	72	56	1	14	—DAC0	—
0x0000_0134	73	57	1	14	MCG	—
0x0000_0138	74	58	1	14	Low Power Timer	—
0x0000_013C	75	59	1	14	Port control module	Pin detect (Port A)
0x0000_0140	76	60	1	15	Port control module	Pin detect (Port B)
0x0000_0144	77	61	1	15	Port control module	Pin detect (Port C)
0x0000_0148	78	62	1	15	Port control module	Pin detect (Port D)
0x0000_014C	79	63	1	15	Port control module	Pin detect (Port E)
0x0000_01B0	80	64	2	16	Software	Software interrupt[4]
0x0000_01B4	81	65	2	16	SPI2	Single interrupt vector for all sources
0x0000_01B8	82	66	2	16	—UART4	—Single interrupt vector for UART status sources
0x0000_01BC	83	67	2	16	—UART4	—Single interrupt vector for UART error sources
0x0000_01C0	84	68	2	17	—UART5	—Single interrupt vector for UART status sources
0x0000_01C4	85	69	2	17	—UART5	—Single interrupt vector for UART error sources
0x0000_01C8	86	70	2	17	—CMP2	—
0x0000_01CC	87	71	2	17	—FTM3	—Single interrupt vector for all sources
0x0000_01D0	88	72	2	18	—DAC1	—
0x0000_01D4	89	73	2	18	—ADC1	—
0x0000_01D8	90	74	2	18	—I^2C2	—
0x0000_01DC	91	75	2	18	—CAN0	—OR'ed Message buffer (0-15)
0x0000_01DC	92	76	2	19	—CAN0	—Bus Off
0x0000_01E0	93	77	2	19	—CAN0	—Error

Table continues on the next page...

Table 15. Interrupt vector assignments (continued)

Address	Vector	IRQ[1]	NVIC non-IPR register number [2]	NVIC IPR register number [3]	Source module	Source description
0x0000_01E4	94	78	2	19	—CAN0	—Transmit Warning
0x0000_01E8	95	79	2	19	—CAN0	—Receive Warning
0x0000_01EC	96	80	2	20	—CAN0	—Wake Up
0x0000_01F0	97	81	2	20	—SDHC	—

1. Indicates the NVIC's interrupt source number.
2. Indicates the NVIC's ISER, ICER, ISPR, ICPR, and IABR register number used for this IRQ. The equation to calculate this value is: IRQ div 32
3. Indicates the NVIC's IPR register number used for this IRQ. The equation to calculate this value is: IRQ div 4
4. This interrupt can only be pended or cleared via the NVIC registers.

Table 16. Interrupt vector assignments

Address	Vector	IRQ[1]	NVIC non-IPR register number [2]	NVIC IPR register number [3]	Source module	Source description
ARM Core System Handler Vectors						
0x0000_0000	0	–	–	–	ARM core	Initial Stack Pointer
0x0000_0004	1	–	–	–	ARM core	Initial Program Counter
0x0000_0008	2	–	–	–	ARM core	Non-maskable Interrupt (NMI)
0x0000_000C	3	–	–	–	ARM core	Hard Fault
0x0000_0010	4	–	–	–	ARM core	MemManage Fault
0x0000_0014	5	–	–	–	ARM core	Bus Fault
0x0000_0018	6	–	–	–	ARM core	Usage Fault
0x0000_001C	7	–	–	–	—	—
0x0000_0020	8	–	–	–	—	—
0x0000_0024	9	–	–	–	—	—
0x0000_0028	10	–	–	–	—	—
0x0000_002C	11	–	–	–	ARM core	Supervisor call (SVCall)
0x0000_0030	12	–	–	–	ARM core	Debug Monitor
0x0000_0034	13	–	–	–	—	—
0x0000_0038	14	–	–	–	ARM core	Pendable request for system service (PendableSrvReq)
0x0000_003C	15	–	–	–	ARM core	System tick timer (SysTick)
Non-Core Vectors						
On-platform Vectors						
0x0000_0040	16	0	0	0	DMA	DMA channel 0, 16 transfer complete
0x0000_0044	17	1	0	0	DMA	DMA channel 1, 17 transfer complete

Table continues on the next page...

Table 16. Interrupt vector assignments (continued)

Address	Vector	IRQ[1]	NVIC non-IPR register number [2]	NVIC IPR register number [3]	Source module	Source description
0x0000_0048	18	2	0	0	DMA	DMA channel 2, 18 transfer complete
0x0000_004C	19	3	0	0	DMA	DMA channel 3, 19 transfer complete
0x0000_0050	20	4	0	1	DMA	DMA channel 4, 20 transfer completeDMA error interrupt channel
0x0000_0054	21	5	0	1	DMA	DMA channel 5, 21 transfer complete–
0x0000_0058	22	6	0	1	DMA Flash memory	DMA channel 6, 22 transfer complete Command complete
0x0000_005C	23	7	0	1	DMAFlash memory	DMA channel 7, 23 transfer completeRead collision
0x0000_0060	24	8	0	2	DMAMode Controller	DMA channel 8, 24 transfer completeLow-voltage detect, low-voltage warning
0x0000_0064	25	9	0	2	DMALLWU	DMA channel 9, 25 transfer complete Low Leakage Wakeup **NOTE:** The LLWU interrupt must not be masked by the interrupt controller to avoid a scenario where the system does not fully exit stop mode on an LLS recovery .
0x0000_0068	26	10	0	2	DMAWDOG	DMA channel 10, 26 transfer completeBoth EWM and WDOG interrupt sources set this IRQ
0x0000_006C	27	11	0	2	DMAI^2C0	DMA channel 11, 27 transfer complete–
0x0000_0070	28	12	0	3	DMASPI0	DMA channel 12, 28 transfer completeSingle interrupt vector for all sources
0x0000_0074	29	13	0	3	DMAI^2S0	DMA channel 13, 29 transfer completeTransmit
0x0000_0078	30	14	0	3	DMAI^2S1	DMA channel 14, 30 transfer completeReceive
0x0000_007C	31	15	0	3	DMAUART0	DMA channel 15, 31 transfer completeSingle interrupt vector for CEA709.1-B (LON) status sources
0x0000_0080	32	16	0	4	DMAUART0	DMA error interrupt channels 0-1531Single interrupt vector for UART status sources
0x0000_0084	33	17	0	4	MCM–UART0	Normal interrupt–Single interrupt vector for UART error sources
Off-platform vectors						
0x0000_0088	34	18	0	4	Flash memoryUART1	Command complete Single interrupt vector for UART status sources

Table continues on the next page...

Table 16. Interrupt vector assignments (continued)

Address	Vector	IRQ[1]	NVIC non-IPR register number [2]	NVIC IPR register number [3]	Source module	Source description
0x0000_008C	35	19	0	4	Flash memoryUART1	Read collisionSingle interrupt vector for UART error sources
0x0000_0090	36	20	0	5	Mode ControllerUART2	Low-voltage detect, low-voltage warningSingle interrupt vector for UART status sources
0x0000_0094	37	21	0	5	LLWUUART2	Low Leakage Wakeup **NOTE:** The LLWU interrupt must not be masked by the interrupt controller to avoid a scenario where the system does not fully exit stop mode on an LLS recovery. Single interrupt vector for UART error sources
0x0000_0098	38	22	0	5	WDOGWDOG or EWMADC0	Watchdog interruptBoth watchdog modules share this interrupt.
0x0000_009C	39	23	0	5	—RNGCMP0	—Randon Number Generator—
0x0000_00A0	40	24	0	6	I^2C0CMP1	—
0x0000_00A4	41	25	0	6	—I^2C1FTM0	—
0x0000_00A8	42	26	0	6	SPI0FTM1	Single interrupt vector for all sources—
0x0000_00AC	43	27	0	6	—SPI1CMT	—Single interrupt vector for all sources —
0x0000_00B0	44	28	0	7	—SPI2RTC	—Single interrupt vector for all sourcesAlarm interrupt
0x0000_00B4	45	29	0	7	—CAN0RTC	—OR'ed Message buffer (0-15)Seconds interrupt
0x0000_00B8	46	30	0	7	—CAN0PIT	—Bus OffChannel 0
0x0000_00BC	47	31	0	7	—CAN0PIT	—ErrorChannel 1
0x0000_00C0	48	32	1	8	—CAN0PIT	—Transmit WarningChannel 2
0x0000_00C4	49	33	1	8	—CAN0PIT	—Receive WarningChannel 3
0x0000_00C8	50	34	1	8	—CAN0PDB	—Wake Up—
0x0000_00CC	51	35	1	8	——I^2S0USB OTGReserved	——Transmit—
0x0000_00D0	52	36	1	9	——I^2S0USB Charger DetectReserved	——Receive—
0x0000_00D4	53	37	1	9	—CAN1TSI	—OR'ed Message buffer (0-15)—
0x0000_00D8	54	38	1	9	—CAN1MCG	—Bus off—
0x0000_00DC	55	39	1	9	—CAN1Low Power Timer	—Error—
0x0000_00E0	56	40	1	10	—CAN1Port control module	—Transmit WarningPin detect (Port A)

Table continues on the next page...

Table 16. Interrupt vector assignments (continued)

Address	Vector	IRQ[1]	NVIC non-IPR register number [2]	NVIC IPR register number [3]	Source module	Source description
0x0000_00E4	57	41	1	10	—CAN1Port control module	—Receive WarningPin detect (Port B)
0x0000_00E8	58	42	1	10	—CAN1Port control module	—Wake UpPin detect (Port C)
0x0000_00EC	59	43	1	10	—Port control module	—Pin detect (Port D)
0x0000_00F0	60	44	1	11	—UART0Port control module	—Single interrupt vector for UART LON sourcesPin detect (Port E)
0x0000_00F4	61	45	1	11	UART0Software initiated interrupt[4]	Single interrupt vector for UART status sources—
0x0000_00F8	62	46	1	11	UART0	Single interrupt vector for UART error sources
0x0000_00FC	63	47	1	11	UART1	Single interrupt vector for UART status sources
0x0000_0100	64	48	1	12	UART1	Single interrupt vector for UART error sources
0x0000_0104	65	49	1	12	UART2	Single interrupt vector for UART status sources
0x0000_0108	66	50	1	12	UART2	Single interrupt vector for UART error sources
0x0000_010C	67	51	1	12	UART3	Single interrupt vector for UART status sources
0x0000_0110	68	52	1	13	UART3	Single interrupt vector for UART error sources
0x0000_0114	69	53	1	13	—UART4	—Single interrupt vector for UART status sources
0x0000_0118	70	54	1	13	—UART4	—Single interrupt vector for UART error sources
0x0000_011C	71	55	1	13	—UART5	—Single interrupt vector for UART status sources
0x0000_0120	72	56	1	14	—UART5	—Single interrupt vector for UART error sources
0x0000_0124	73	57	1	14	ADC0	—
0x0000_0128	74	58	1	14	—ADC1	—
0x0000_012C	75	59	1	14	CMP0	—
0x0000_0130	76	60	1	15	—CMP1	—
0x0000_0134	77	61	1	15	—CMP2	—
0x0000_0138	78	62	1	15	FTM0	Single interrupt vector for all sources
0x0000_013C	79	63	1	15	—FTM1	—Single interrupt vector for all sources
0x0000_0140	80	64	2	16	—FTM2	—Single interrupt vector for all sources
0x0000_0144	81	65	2	16	CMT	—

Table continues on the next page...

Table 16. Interrupt vector assignments (continued)

Address	Vector	IRQ[1]	NVIC non-IPR register number [2]	NVIC IPR register number [3]	Source module	Source description
0x0000_0148	82	66	2	16	RTC	Alarm interrupt
0x0000_014C	83	67	2	16	—RTC	—Seconds interrupt
0x0000_0150	84	68	2	17	PIT	Channel 0
0x0000_0154	85	69	2	17	PIT	Channel 1
0x0000_0158	86	70	2	17	PIT	Channel 2
0x0000_015C	87	71	2	17	PIT	Channel 3
0x0000_0160	88	72	2	18	PDB	—
0x0000_0164	89	73	2	18	—USB OTG	—
0x0000_0168	90	74	2	18	—USB Charger Detect	—
0x0000_016C	91	75	2	18	—Ethernet MAC	—IEEE 1588 Timer Interrupt
0x0000_0170	92	76	2	19	—Ethernet MAC	—Transmit interrupt
0x0000_0174	93	77	2	19	—Ethernet MAC	—Receive interrupt
0x0000_0178	94	78	2	19	—Ethernet MAC	—Error and miscellaneous interrupt
0x0000_017C	95	79	2	19	—I^2S0	—
0x0000_0180	96	80	2	20	—SDHC	—
0x0000_0184	97	81	2	20	—DAC0	—
0x0000_0188	98	82	2	20	—DAC1	—
0x0000_018C	99	83	2	20	TSI	Single interrupt vector for all sources
0x0000_0190	100	84	2	21	MCG	—
0x0000_0194	101	85	2	21	Low Power Timer	—
0x0000_0198	102	86	2	21	—Segment LCD	—Single interrupt vector for all sources
0x0000_019C	103	87	2	21	Port control module	Pin detect (Port A)
0x0000_01A0	104	88	2	22	Port control module	Pin detect (Port B)
0x0000_01A4	105	89	2	22	Port control module	Pin detect (Port C)
0x0000_01A8	106	90	2	22	Port control module	Pin detect (Port D)
0x0000_01AC	107	91	2	22	Port control module	Pin detect (Port E)
0x0000_01B0	108	92	2	23	—Port control module	—Pin detect (Port F)
0x0000_01B4	109	93	2	23	—DDR controller	—
0x0000_01B8	110	94	2	23	Software	Software interrupt[5]
0x0000_01BC	111	95	2	23	—NAND flash controller (NFC)	—
0x0000_01C0	112	96	3	24	—USB HS	—
0x0000_01C4	113	97	3	24	—Graphical LCD	—
0x0000_01C8	114	98	3	24	—CMP3	—
0x0000_01CC	115	99	3	24	—Tamper	—DryIce Tamper Interrupt

Table continues on the next page...

Table 16. Interrupt vector assignments (continued)

Address	Vector	IRQ[1]	NVIC non-IPR register number [2]	NVIC IPR register number [3]	Source module	Source description
0x0000_01D0	116	100	3	25	—	—
0x0000_01D4	117	101	3	25	—FTM3	—Single interrupt vector for all sources
0x0000_01D8	118	102	3	25	—ADC2	—
0x0000_01DC	119	103	3	25	—ADC3	—
0x0000_01E0	120	104	3	26	—I^2S1	—Transmit
0x0000_01E4	121	105	3	26	—I^2S1	—Receive

1. Indicates the NVIC's interrupt source number.
2. Indicates the NVIC's ISER, ICER, ISPR, ICPR, and IABR register number used for this IRQ. The equation to calculate this value is: IRQ div 32
3. Indicates the NVIC's IPR register number used for this IRQ. The equation to calculate this value is: IRQ div 4
4. This interrupt can only be pended or cleared via the NVIC registers.
5. This interrupt can only be pended or cleared via the NVIC registers.

Table 17. Interrupt vector assignments

Address	Vector	IRQ[1]	NVIC non-IPR register number [2]	NVIC IPR register number [3]	Source module	Source description
ARM Core System Handler Vectors						
0x0000_0000	0	–	–	–	ARM core	Initial Stack Pointer
0x0000_0004	1	–	–	–	ARM core	Initial Program Counter
0x0000_0008	2	–	–	–	ARM core	Non-maskable Interrupt (NMI)
0x0000_000C	3	–	–	–	ARM core	Hard Fault
0x0000_0010	4	–	–	–	ARM core	MemManage Fault
0x0000_0014	5	–	–	–	ARM core	Bus Fault
0x0000_0018	6	–	–	–	ARM core	Usage Fault
0x0000_001C	7	–	–	–	—	—
0x0000_0020	8	–	–	–	—	—
0x0000_0024	9	–	–	–	—	—
0x0000_0028	10	–	–	–	—	—
0x0000_002C	11	–	–	–	ARM core	Supervisor call (SVCall)
0x0000_0030	12	–	–	–	ARM core	Debug Monitor
0x0000_0034	13	–	–	–	—	—
0x0000_0038	14	–	–	–	ARM core	Pendable request for system service (PendableSrvReq)
0x0000_003C	15	–	–	–	ARM core	System tick timer (SysTick)
Non-Core Vectors						
On-platform Vectors						

Table continues on the next page...

Table 17. Interrupt vector assignments (continued)

Address	Vector	IRQ[1]	NVIC non-IPR register number [2]	NVIC IPR register number [3]	Source module	Source description
0x0000_0040	16	0	0	0	DMA	DMA channel 0, 16 transfer complete
0x0000_0044	17	1	0	0	DMA	DMA channel 1, 17 transfer complete
0x0000_0048	18	2	0	0	DMA	DMA channel 2, 18 transfer complete
0x0000_004C	19	3	0	0	DMA	DMA channel 3, 19 transfer complete
0x0000_0050	20	4	0	1	DMA	DMA channel 4, 20 transfer complete
0x0000_0054	21	5	0	1	DMA	DMA channel 5, 21 transfer complete
0x0000_0058	22	6	0	1	DMA	DMA channel 6, 22 transfer complete
0x0000_005C	23	7	0	1	DMA	DMA channel 7, 23 transfer complete
0x0000_0060	24	8	0	2	DMA	DMA channel 8, 24 transfer complete
0x0000_0064	25	9	0	2	DMA	DMA channel 9, 25 transfer complete
0x0000_0068	26	10	0	2	DMA	DMA channel 10, 26 transfer complete
0x0000_006C	27	11	0	2	DMA	DMA channel 11, 27 transfer complete
0x0000_0070	28	12	0	3	DMA	DMA channel 12, 28 transfer complete
0x0000_0074	29	13	0	3	DMA	DMA channel 13, 29 transfer complete
0x0000_0078	30	14	0	3	DMA	DMA channel 14, 30 transfer complete
0x0000_007C	31	15	0	3	DMA	DMA channel 15, 31 transfer complete
0x0000_0080	32	16	0	4	DMA	DMA error interrupt channels 0-1531
0x0000_0084	33	17	0	4	MCM	Normal interrupt
Off-platform vectors						
0x0000_0088	34	18	0	4	Flash memory	Command complete
0x0000_008C	35	19	0	4	Flash memory	Read collision
0x0000_0090	36	20	0	5	Mode Controller	Low-voltage detect, low-voltage warning
0x0000_0094	37	21	0	5	LLWU	Low Leakage Wakeup **NOTE:** The LLWU interrupt must not be masked by the interrupt controller to avoid a scenario where the system does not fully exit stop mode on an LLS recovery.
0x0000_0098	38	22	0	5	WDOG	—
0x0000_009C	39	23	0	5	—RNG	—Randon Number Generator
0x0000_00A0	40	24	0	6	I^2C0	—
0x0000_00A4	41	25	0	6	—I^2C1	—
0x0000_00A8	42	26	0	6	SPI0	Single interrupt vector for all sources
0x0000_00AC	43	27	0	6	—SPI1	—Single interrupt vector for all sources
0x0000_00B0	44	28	0	7	—SPI2	—Single interrupt vector for all sources
0x0000_00B4	45	29	0	7	—CAN0	—OR'ed Message buffer (0-15)
0x0000_00B8	46	30	0	7	—CAN0	—Bus Off

Table continues on the next page...

Table 17. Interrupt vector assignments (continued)

Address	Vector	IRQ[1]	NVIC non-IPR register number [2]	NVIC IPR register number [3]	Source module	Source description
0x0000_00BC	47	31	0	7	—CAN0	—Error
0x0000_00C0	48	32	1	8	—CAN0	—Transmit Warning
0x0000_00C4	49	33	1	8	—CAN0	—Receive Warning
0x0000_00C8	50	34	1	8	—CAN0	—Wake Up
0x0000_00CC	51	35	1	8	——I^2S0	——Transmit
0x0000_00D0	52	36	1	9	——I^2S0	——Receive
0x0000_00D4	53	37	1	9	—CAN1	—OR'ed Message buffer (0-15)
0x0000_00D8	54	38	1	9	—CAN1	—Bus off
0x0000_00DC	55	39	1	9	—CAN1	—Error
0x0000_00E0	56	40	1	10	—CAN1	—Transmit Warning
0x0000_00E4	57	41	1	10	—CAN1	—Receive Warning
0x0000_00E8	58	42	1	10	—CAN1	—Wake Up
0x0000_00EC	59	43	1	10	—	—
0x0000_00F0	60	44	1	11	—UART0	—Single interrupt vector for UART LON sources
0x0000_00F4	61	45	1	11	UART0	Single interrupt vector for UART status sources
0x0000_00F8	62	46	1	11	UART0	Single interrupt vector for UART error sources
0x0000_00FC	63	47	1	11	UART1	Single interrupt vector for UART status sources
0x0000_0100	64	48	1	12	UART1	Single interrupt vector for UART error sources
0x0000_0104	65	49	1	12	UART2	Single interrupt vector for UART status sources
0x0000_0108	66	50	1	12	UART2	Single interrupt vector for UART error sources
0x0000_010C	67	51	1	12	UART3	Single interrupt vector for UART status sources
0x0000_0110	68	52	1	13	UART3	Single interrupt vector for UART error sources
0x0000_0114	69	53	1	13	—UART4	—Single interrupt vector for UART status sources
0x0000_0118	70	54	1	13	—UART4	—Single interrupt vector for UART error sources
0x0000_011C	71	55	1	13	—UART5	—Single interrupt vector for UART status sources
0x0000_0120	72	56	1	14	—UART5	—Single interrupt vector for UART error sources
0x0000_0124	73	57	1	14	ADC0	—

Table continues on the next page...

Table 17. Interrupt vector assignments (continued)

Address	Vector	IRQ[1]	NVIC non-IPR register number [2]	NVIC IPR register number [3]	Source module	Source description
0x0000_0128	74	58	1	14	—ADC1	—
0x0000_012C	75	59	1	14	CMP0	—
0x0000_0130	76	60	1	15	—CMP1	—
0x0000_0134	77	61	1	15	—CMP2	—
0x0000_0138	78	62	1	15	FTM0	Single interrupt vector for all sources
0x0000_013C	79	63	1	15	—FTM1	—Single interrupt vector for all sources
0x0000_0140	80	64	2	16	—FTM2	—Single interrupt vector for all sources
0x0000_0144	81	65	2	16	CMT	—
0x0000_0148	82	66	2	16	RTC	Alarm interrupt
0x0000_014C	83	67	2	16	—RTC	—Seconds interrupt
0x0000_0150	84	68	2	17	PIT	Channel 0
0x0000_0154	85	69	2	17	PIT	Channel 1
0x0000_0158	86	70	2	17	PIT	Channel 2
0x0000_015C	87	71	2	17	PIT	Channel 3
0x0000_0160	88	72	2	18	PDB	—
0x0000_0164	89	73	2	18	—USB OTG	—
0x0000_0168	90	74	2	18	—USB Charger Detect	—
0x0000_016C	91	75	2	18	—Ethernet MAC	—IEEE 1588 Timer Interrupt
0x0000_0170	92	76	2	19	—Ethernet MAC	—Transmit interrupt
0x0000_0174	93	77	2	19	—Ethernet MAC	—Receive interrupt
0x0000_0178	94	78	2	19	—Ethernet MAC	—Error and miscellaneous interrupt
0x0000_017C	95	79	2	19	—I^2S0	—
0x0000_0180	96	80	2	20	—SDHC	—
0x0000_0184	97	81	2	20	—DAC0	—
0x0000_0188	98	82	2	20	—DAC1	—
0x0000_018C	99	83	2	20	TSI	Single interrupt vector for all sources
0x0000_0190	100	84	2	21	MCG	—
0x0000_0194	101	85	2	21	Low Power Timer	—
0x0000_0198	102	86	2	21	—Segment LCD	—Single interrupt vector for all sources
0x0000_019C	103	87	2	21	Port control module	Pin detect (Port A)
0x0000_01A0	104	88	2	22	Port control module	Pin detect (Port B)
0x0000_01A4	105	89	2	22	Port control module	Pin detect (Port C)
0x0000_01A8	106	90	2	22	Port control module	Pin detect (Port D)
0x0000_01AC	107	91	2	22	Port control module	Pin detect (Port E)
0x0000_01B0	108	92	2	23	—Port control module	—Pin detect (Port F)

Table continues on the next page...

Table 17. Interrupt vector assignments (continued)

Address	Vector	IRQ[1]	NVIC non-IPR register number [2]	NVIC IPR register number [3]	Source module	Source description
0x0000_01B4	109	93	2	23	—DDR controller	—
0x0000_01B8	110	94	2	23	Software	Software interrupt[4]
0x0000_01BC	111	95	2	23	—NAND flash controller (NFC)	—
0x0000_01C0	112	96	3	24	—USB HS	—
0x0000_01C4	113	97	3	24	—Graphical LCD	—
0x0000_01C8	114	98	3	24	—CMP3	—
0x0000_01CC	115	99	3	24	—Tamper	—DryIce Tamper Interrupt
0x0000_01D0	116	100	3	25	—	—
0x0000_01D4	117	101	3	25	—FTM3	—Single interrupt vector for all sources
0x0000_01D8	118	102	3	25	—ADC2	—
0x0000_01DC	119	103	3	25	—ADC3	—
0x0000_01E0	120	104	3	26	—I^2S1	—Transmit
0x0000_01E4	121	105	3	26	—I^2S1	—Receive

1. Indicates the NVIC's interrupt source number.
2. Indicates the NVIC's ISER, ICER, ISPR, ICPR, and IABR register number used for this IRQ. The equation to calculate this value is: IRQ div 32
3. Indicates the NVIC's IPR register number used for this IRQ. The equation to calculate this value is: IRQ div 4
4. This interrupt can only be pended or cleared via the NVIC registers.

Table 18. Interrupt vector assignments

Address	Vector	IRQ[1]	NVIC non-IPR register number [2]	NVIC IPR register number [3]	Source module	Source description
ARM Core System Handler Vectors						
0x0000_0000	0	–	–	–	ARM core	Initial Stack Pointer
0x0000_0004	1	–	–	–	ARM core	Initial Program Counter
0x0000_0008	2	–	–	–	ARM core	Non-maskable Interrupt (NMI)
0x0000_000C	3	–	–	–	ARM core	Hard Fault
0x0000_0010	4	–	–	–	ARM core	MemManage Fault
0x0000_0014	5	–	–	–	ARM core	Bus Fault
0x0000_0018	6	–	–	–	ARM core	Usage Fault
0x0000_001C	7	–	–	–	—	—
0x0000_0020	8	–	–	–	—	—
0x0000_0024	9	–	–	–	—	—
0x0000_0028	10	–	–	–	—	—

Table continues on the next page...

Table 18. Interrupt vector assignments (continued)

Address	Vector	IRQ[1]	NVIC non-IPR register number [2]	NVIC IPR register number [3]	Source module	Source description
0x0000_002C	11	–	–	–	ARM core	Supervisor call (SVCall)
0x0000_0030	12	–	–	–	ARM core	Debug Monitor
0x0000_0034	13	–	–	–	—	—
0x0000_0038	14	–	–	–	ARM core	Pendable request for system service (PendableSrvReq)
0x0000_003C	15	–	–	–	ARM core	System tick timer (SysTick)
Non-Core Vectors						
On-platform Vectors						
0x0000_0040	16	0	0	0	DMA	DMA channel 0, 16 transfer complete
0x0000_0044	17	1	0	0	DMA	DMA channel 1, 17 transfer complete
0x0000_0048	18	2	0	0	DMA	DMA channel 2, 18 transfer complete
0x0000_004C	19	3	0	0	DMA	DMA channel 3, 19 transfer complete
0x0000_0050	20	4	0	1	DMA	DMA channel 4, 20 transfer complete
0x0000_0054	21	5	0	1	DMA	DMA channel 5, 21 transfer complete
0x0000_0058	22	6	0	1	DMA	DMA channel 6, 22 transfer complete
0x0000_005C	23	7	0	1	DMA	DMA channel 7, 23 transfer complete
0x0000_0060	24	8	0	2	DMA	DMA channel 8, 24 transfer complete
0x0000_0064	25	9	0	2	DMA	DMA channel 9, 25 transfer complete
0x0000_0068	26	10	0	2	DMA	DMA channel 10, 26 transfer complete
0x0000_006C	27	11	0	2	DMA	DMA channel 11, 27 transfer complete
0x0000_0070	28	12	0	3	DMA	DMA channel 12, 28 transfer complete
0x0000_0074	29	13	0	3	DMA	DMA channel 13, 29 transfer complete
0x0000_0078	30	14	0	3	DMA	DMA channel 14, 30 transfer complete
0x0000_007C	31	15	0	3	DMA	DMA channel 15, 31 transfer complete
0x0000_0080	32	16	0	4	DMA	DMA error interrupt channels 0-15
0x0000_0084	33	17	0	4	MCM	—
Off-platform vectors						
0x0000_0088	34	18	0	4	Flash memory	Command complete
0x0000_008C	35	19	0	4	Flash memory	Read collision
0x0000_0090	36	20	0	5	Mode Controller	Low-voltage detect, low-voltage warning
0x0000_0094	37	21	0	5	LLWU	Low Leakage Wakeup **NOTE:** The LLWU interrupt must not be masked by the interrupt controller to avoid a scenario where the system does not fully exit stop mode on an LLS recovery.
0x0000_0098	38	22	0	5	WDOG	—

Table continues on the next page...

Table 18. Interrupt vector assignments (continued)

Address	Vector	IRQ[1]	NVIC non-IPR register number [2]	NVIC IPR register number [3]	Source module	Source description
0x0000_009C	39	23	0	5	—RNG	—Randon Number Generator
0x0000_00A0	40	24	0	6	I^2C0	—
0x0000_00A4	41	25	0	6	—I^2C1	—
0x0000_00A8	42	26	0	6	SPI0	Single interrupt vector for all sources
0x0000_00AC	43	27	0	6	—SPI1	—Single interrupt vector for all sources
0x0000_00B0	44	28	0	7	—I^2S0	Transmit
0x0000_00B4	45	29	0	7	—I^2S0	Receive
0x0000_00B8	46	30	0	7	UART0	Single interrupt vector for UART LON sources
0x0000_00BC	47	31	0	7	UART0	Single interrupt vector for UART status sources
0x0000_00C0	48	32	1	8	UART0	Single interrupt vector for UART error sources
0x0000_00C4	49	33	1	8	UART1	Single interrupt vector for UART status sources
0x0000_00C8	50	34	1	8	UART1	Single interrupt vector for UART error sources
0x0000_00CC	51	35	1	8	UART2	Single interrupt vector for UART status sources
0x0000_00D0	52	36	1	9	UART2	Single interrupt vector for UART error sources
0x0000_00D4	53	37	1	9	UART3	Single interrupt vector for UART status sources
0x0000_00D8	54	38	1	9	UART3	Single interrupt vector for UART error sources
0x0000_00DC	55	39	1	9	ADC0	—
0x0000_00E0	56	40	1	10	CMP0	—
0x0000_00E4	57	41	1	10	CMP1	—
0x0000_00E8	58	42	1	10	FTM0	Single interrupt vector for all sources
0x0000_00EC	59	43	1	10	—FTM1	—Single interrupt vector for all sources
0x0000_00F0	60	44	1	11	—FTM2	—Single interrupt vector for all sources
0x0000_00F4	61	45	1	11	CMT	—
0x0000_00F8	62	46	1	11	RTC	Alarm interrupt
0x0000_00FC	63	47	1	11	RTC	Seconds interrupt
0x0000_0100	64	48	1	12	PIT	Channel 0
0x0000_0104	65	49	1	12	PIT	Channel 1
0x0000_0108	66	50	1	12	PIT	Channel 2
0x0000_010C	67	51	1	12	PIT	Channel 3
0x0000_0110	68	52	1	13	PDB	—

Table continues on the next page...

Table 18. Interrupt vector assignments (continued)

Address	Vector	IRQ[1]	NVIC non-IPR register number [2]	NVIC IPR register number [3]	Source module	Source description
0x0000_0114	69	53	1	13	—USB OTG	—
0x0000_0118	70	54	1	13	—USB Charger Detect	—
0x0000_011C	71	55	1	13	—DryIce	—Tamper
0x0000_0120	72	56	1	14	—DAC0	—
0x0000_0124	73	57	1	14	MCG	—
0x0000_0128	74	58	1	14	Low Power Timer	—
0x0000_012C	75	59	1	14	Port control module	Pin detect (Port A)
0x0000_0130	76	60	1	15	Port control module	Pin detect (Port B)
0x0000_0134	77	61	1	15	Port control module	Pin detect (Port C)
0x0000_0138	78	62	1	15	Port control module	Pin detect (Port D)
0x0000_013C	79	63	1	15	Port control module	Pin detect (Port E)
0x0000_0140	80	64	2	16	Software	Software interrupt[4]

1. Indicates the NVIC's interrupt source number.
2. Indicates the NVIC's ISER, ICER, ISPR, ICPR, and IABR register number used for this IRQ. The equation to calculate this value is: IRQ div 32
3. Indicates the NVIC's IPR register number used for this IRQ. The equation to calculate this value is: IRQ div 4
4. This interrupt can only be pended or cleared via the NVIC registers.

6 Revision history

Revision	Date	Description of changes
0	2/2012	Initial release.
1	8/2012	Included "Mode Controller Version 1 to Version 2" section.

How to Reach Us:

Home Page:
www.freescale.com

Web Support:
http://www.freescale.com/support

USA/Europe or Locations Not Listed:
Freescale Semiconductor
Technical Information Center, EL516
2100 East Elliot Road
Tempe, Arizona 85284
+1-800-521-6274 or +1-480-768-2130
www.freescale.com/support

Europe, Middle East, and Africa:
Freescale Halbleiter Deutschland GmbH
Technical Information Center
Schatzbogen 7
81829 Muenchen, Germany
+44 1296 380 456 (English)
+46 8 52200080 (English)
+49 89 92103 559 (German)
+33 1 69 35 48 48 (French)
www.freescale.com/support

Japan:
Freescale Semiconductor Japan Ltd.
Headquarters
ARCO Tower 15F
1-8-1, Shimo-Meguro, Meguro-ku,
Tokyo 153-0064
Japan
0120 191014 or +81 3 5437 9125
support.japan@freescale.com

Asia/Pacific:
Freescale Semiconductor China Ltd.
Exchange Building 23F
No. 118 Jianguo Road
Chaoyang District
Beijing 100022
China
+86 10 5879 8000
support.asia@freescale.com

Document Number: AN4445
Rev. 1, 8/2012

freescale